



Escola d'Enginyeria de Telecomunicació i  
Aeroespacial de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

# TRABAJO FINAL DE GRADO

**TÍTULO DEL TFG:** Gestor documental para la documentación oficial de una compañía

**TITULACIÓN:** Grau en Enginyeria Telemàtica

**AUTOR:** Sergi Galceran Serrano

**DIRECTOR:** Toni Oller

**DATA:** 9 de julio del 2019



**Título:** Gestor documental para la documentación oficial de una compañía

**Autor:** Sergi Galceran Serrano

**Director:** Toni Oller

**Data:** 23 de julio del 2019

## **Resumen**

Actualmente ha habido un aumento en el uso de software relacionado con el archivo digital. Es una tendencia que está en crecimiento, y cada vez se pueden encontrar más empresas que deciden invertir en este tipo de softwares. Esto permite que haya una gestión mucho más rápida y ágil de los documentos. También permite tener un acceso mucho más controlado y disponible para todo el mundo de los documentos.

En este proyecto se explicará el proceso que se ha seguido para el desarrollo de un gestor documental para una compañía. Este gestor documental permitirá la subida de los documentos oficiales de la compañía. En el gestor se podrán subir documentos que podrán ser restringidos, o no, a un determinado grupo de usuarios. Se podrán subir documentos que requieran la aceptación de leído y comprendido por parte de los usuarios que la persona que suba el documento indique.

**Title:** Documentary manager for the documentation of a company

**Author:** Sergi Galceran Serrano

**Director:** Toni Oller

**Date:** July 9th 2019

## Overview

There has been an increase in the use of software related to the digital archive. It is a trend that is growing, and every time you can find more companies that decide to invest in this type of software. This allows for a much faster and more agile management of documents. It also allows for much more controlled and available access to the world of documents.

In this project the process that has been followed for the development of a document manager for a company will be explained. This document manager will allow the upload of the official documents of the company. In the manager you can upload documents that may be restricted, or not, to a certain group of users. You can upload documents that require the acceptance of read and understood by the users that the person that uploads the document indicates.

# ÍNDICE

<b>Introducción.....</b>	<b>10</b>
<b>CAPÍTULO 1. OBJETIVOS PRINCIPALES .....</b>	<b>11</b>
1.1. Objetivo del proyecto .....	11
1.2. Situación actual .....	11
1.3. Requerimientos básicos .....	12
1.3.1. Documentos Controlados .....	12
1.3.2. Documentos Internos .....	13
1.4. Solución ofrecida.....	13
<b>CAPÍTULO 2. PLATAFORMA DE LA SOLUCIÓN .....</b>	<b>14</b>
2.1 Sharepoint .....	14
2.2 Historia .....	14
2.2.1 SharePoint Portal Server 2001 .....	14
2.2.2 SharePoint Portal Server 2003.....	15
2.2.3 Office SharePoint Server 2007.....	16
2.2.4 SharePoint Server 2010 .....	16
2.2.5 SharePoint Server 2013 .....	17
2.2.6 SharePoint Server 2016 .....	17
2.2.7 SharePoint Online – Office 365 .....	18
<b>CAPÍTULO 3. ARQUITECTURA .....</b>	<b>19</b>
3.1. Modelo de objetos .....	19
3.1.1 Granja .....	19
3.1.2 Base de datos de contenidos .....	19
3.1.3 Aplicación Web .....	19
3.1.4 Servicios .....	19
3.1.5 Colección de sitios.....	20
3.1.6 Sitio .....	20
3.1.7 Lista/Biblioteca .....	20
3.1.8 Modelo de datos .....	20
3.1.9 Listas .....	20
3.1.10 Columnas.....	21
3.1.11 Tipos de contenido .....	21
3.1.12 Seguridad y permisos .....	21
<b>CAPÍTULO 4. DESARROLLO DE LA SOLUCIÓN .....</b>	<b>22</b>
4.1 Herramientas utilizadas .....	22
4.1.1 Visual Studio.....	22
4.2 Diseño de la solución.....	22
4.2.1 Funcionalidades de la solución .....	22
4.2.2 Diseño conceptual .....	23
4.3 Diseño físico .....	28

4.3.1	Fases de la solución .....	28
<b>4.4</b>	<b>Creación del site de SharePoint.....</b>	<b>28</b>
4.4.1	Programación de la solución .....	30
<b>4.5</b>	<b>Creación de la solución .....</b>	<b>31</b>
4.5.1	Columnas de sitio .....	31
4.5.2	Contenidos del sitio .....	33
4.5.3	Listas y bibliotecas .....	34
4.5.4	Receptores de evento .....	35
4.5.5	Receptor documentos controlados.....	37
4.5.6	Receptor documentos internos.....	39
4.5.7	Timer Jobs .....	39
4.5.8	Timer Job de Caducidad de Documentos .....	42
4.5.9	Timer Job de Envío de Correos.....	43
4.5.10	Timer Job de Recordatorios .....	44
4.5.11	Estructura final del proyecto e instalación.....	46
<b>CAPÍTULO 5.</b>	<b>SOLUCIÓN FINAL .....</b>	<b>50</b>
<b>5.1</b>	<b>Documentos internos.....</b>	<b>50</b>
<b>5.2</b>	<b>Documentos controlados .....</b>	<b>52</b>
<b>5.3</b>	<b>Listas de configuración .....</b>	<b>54</b>
<b>5.4</b>	<b>Caducidad de documentos.....</b>	<b>56</b>
<b>5.5</b>	<b>Leídos y comprendidos .....</b>	<b>56</b>
<b>CAPÍTULO 6.</b>	<b>CONCLUSIONES.....</b>	<b>60</b>
<b>Bibliografía.....</b>		<b>61</b>

## ÍNDICE DE FIGURAS

Ilustración 2.1. Pantalla SharePoint Portal Server 2001 .....	153
Ilustración 2.2 Pantalla SharePoint Portal Server 2003.....	154
Ilustración 2.3. Pantalla Office SharePoint Server 2007 .....	164
Ilustración 2.4. Pantalla Office SharePoint Server 2010 .....	175
Ilustración 2.5. Pantalla Office SharePoint Server 2013 .....	176
Ilustración 2.6. Pantalla SharePoint Online .....	187
Ilustración 3.1. Esquema de servidores .....	189
Ilustración 4.1 Proceso publicación de documentos.....	22
Ilustración 4.2 Flujo de funcionamiento para la lista de estado.....	25
Ilustración 4.3 Flujo formulario leído y comprendido.....	25
Ilustración 4.4 Flujo funcionamiento restringir documentos.....	26
Ilustración 4.5 Proceso nuevas incorporaciones.....	26
Ilustración 4.6 Proceso de recordatorios.....	26
Ilustración 4.7 Pantalla central admin.....	28
Ilustración 4.8 Pantalla creación de site.....	29
Ilustración 4.9 Formulario creación nuevo ítem .....	31
Ilustración 4.10 Estructura del elemento en formato xml .....	32
Ilustración 4.11 Tabla de Contenido de Sitio.....	33
Ilustración 4.12 Formulario creación de lista .....	34
Ilustración 4.13 Configuración columnas de la lista .....	34
Ilustración 4.14 Formulario creación receptores de eventos .....	36
Ilustración 4.15 Código lógica añadir código de documento.....	37
Ilustración 4.16 Código asignación permisos documento .....	37
Ilustración 4.17 Código poner en cola un ítem .....	37
Ilustración 4.18 Estructura de un TimerJob.....	38
Ilustración 4.19 Código activación del TimerJob .....	39
Ilustración 4.20 Código creación TimerJob.....	40
Ilustración 4.21 Código eliminación TimerJob existente .....	40
Ilustración 4.22 Código obtención días de la lista de configuración .....	41
Ilustración 4.23 Código obtener documentos caducados .....	41
Ilustración 4.24 CamlQuery obtención de documentos activos y caducados .....	42
Ilustración 4.25 Código obtención configuración del correo .....	42
Ilustración 4.26 Definición del TimerJob Worker.....	43
Ilustración 4.27 Código WorkItem de envío de correos .....	44
Ilustración 4.28 Obtención valores de la lista de configuración .....	44
Ilustración 4.29 Código envío correos del Worker .....	44
Ilustración 4.30 Estructura feature .....	45
Ilustración 4.31 Pantalla de la feature .....	46
Ilustración 4.32 Javascript desactivación link items .....	47
Ilustración 4.33 Javascript redirección al formulario custom .....	47
Ilustración 4.34 Pagina en modo edición .....	48
Ilustración 4.35 Formulario edición de la webpart .....	48
Ilustración 5.1 Biblioteca Documentos Internos.....	50
Ilustración 5.2 Formulario carga documento.....	50
Ilustración 5.3 Formulario propiedades del documento.....	51
Ilustración 5.4 Error al no rellenar los campos .....	52
Ilustración 5.5 Biblioteca documentos controlados .....	52
Ilustración 5.6 Formulario propiedades documentos controlados .....	53
Ilustración 5.7 Error al no rellenar los campos.....	53
Ilustración 5.8 Lista configuración recordatorios .....	54
Ilustración 5.9 Lista configuración correos .....	54
Ilustración 5.10 Lista configuración departamentos.....	55
Ilustración 5.11 Lista configuración de empresas .....	55
Ilustración 5.12 Lista configuración idioma .....	56
Ilustración 5.13 Correo caducidad documento.....	56
Ilustración 5.14 Correo nuevo documento leído y comprendido .....	57
Ilustración 5.15 Formulario leído y comprendido sin aceptar.....	57

Ilustración 5.16 Formulario leído y comprendido aceptado .....	58
Ilustración 5.17 Lista estado documentos .....	58
Ilustración 5.18 Correo recordatorio lectura documento .....	59



## GLOSARIO

**UI** *User Interface*, es el medio con que el usuario puede comunicarse con una máquina, equipo, computadora o dispositivo, y comprende todos los puntos de contacto entre el usuario y el equipo.

**IIS** *Internet Information Services*, es un servidor web y un conjunto de servicios para el sistema operativo Microsoft Windows.

**XML** *eXtensible Markup Language*, es un lenguaje de marcado similar a HTML.

## INTRODUCCIÓN

Este trabajo tiene como objetivo el desarrollo de un gestor documental de una compañía. En el primer capítulo se presentan los objetivos del proyecto, así como también la situación que hay actualmente. Por último, se explicarán los requerimientos básicos que se pretenden conseguir con el proyecto, así como la solución ofrecida.

En el segundo capítulo se hablará sobre la solución escogida, hablando sobre su historia y su evolución.

En el tercer apartado se presenta la arquitectura que presenta la solución que se ha escogido, hablando del modelo de objetos que la compone.

En el siguiente apartado se centra en el desarrollo de la solución, presentando las herramientas utilizadas en el desarrollo de la solución, así como su diseño.

En este apartado también se habla sobre cómo se ha desarrollado la estructura de la solución y como se ha implementado en el cliente.

Antes de concluir en el quinto apartado se muestran los resultados de la solución ya implementada en el cliente y el funcionamiento de esta.

Para finalizar, en el último apartado se encuentran unas conclusiones sobre el proyecto.

## **CAPÍTULO 1. OBJETIVOS PRINCIPALES**

En este apartado se presentarán los objetivos del proyecto, cual es la situación actual del cliente y los requerimientos básicos que se pretenden alcanzar.

### **1.1. Objetivo del proyecto**

El objetivo principal del proyecto es disponer de un sistema para el almacenamiento de la documentación oficial de la compañía, el cual sea el acceso único a dicha documentación.

Como único punto de acceso a la documentación oficial de la compañía deberá permitir a los empleados localizar y consultar la documentación que haya sido almacenada de forma rápida y fácil.

Al tratarse de documentación sensible, en caso necesario también se podrá dar un acceso restringido, a determinados documentos, a un determinado grupo de usuarios de la compañía.

Finalmente, el sistema ha de dar trazabilidad y permitir demostrar si los usuarios han leído un documento y han aceptado su contenido, para de esta forma poder demostrar ante una auditoría el seguimiento de este proceso.

### **1.2. Situación actual**

Actualmente el cliente no dispone de un único lugar donde se ubique toda la documentación oficial de la compañía y cada departamento la pone a disposición de los empleados de una manera distinta provocando que si un empleado quiere consultar toda la documentación oficial no sea un proceso ágil ni se puede garantizar y demostrar que se realice.

Para el seguimiento y el proceso de aceptación de los documentos de lectura obligatoria, actualmente se realiza manualmente a papel, con un proceso manual de firma o vía correo electrónico.

La búsqueda de documentos, en algunos casos, resulta engorrosa, y localizar un determinado Documento no siempre tiene la garantía de que se está tratando la última versión de este, es decir la versión vigente.

### 1.3. Requerimientos básicos

Los requerimientos básicos definidos con el cliente y que deben ser cubiertos por el proyecto son:

- Que un documento, en su versión oficial, se encuentre en una única ubicación dentro de la organización, siendo, por política de empresa, la única ubicación oficial donde se podrá alojar.
- Que se disponga de un control para garantizar que se está poniendo a disposición de los empleados la última versión de un determinado Documento, de manera que, todo documento publicado en el espacio de documentos será documento oficial y validado previamente por los promotores de dicho documento.
- Que se garantice que no exista más de una versión para un determinado Documento en distintos archivos en un estado vigente. Esto implica que cuando de un Documento se publica una nueva versión, la anterior ha de dejar de estar vigente.

La tipología de los Documentos a almacenar se puede distinguir en función del “ciclo de vida” del Documento, de las acciones que se pueden realizar sobre el mismo, y de las reglas que rigen su gestión que actuarán de forma similar.

Los dos tipos de Documentos a tener en cuenta son los siguientes:

#### 1.3.1. Documentos Controlados

Documentos que precisan una gestión especial, con el objetivo de cumplir con las normativas legales y/o jurídicas.

- Para este tipo de Documento, se podrá realizar el registro de “Leído y comprendido”, para un grupo cerrado y/o uno o varios usuarios específicos.
- Se dejará el Sistema preparado para que la distribución del Documento pueda ser para TODA la plantilla de la compañía, un grupo de personas y/o un usuario concreto.
- Por otro lado, la Autorización para la consulta del Documento será configurada a nivel de Documento. En determinados casos, un Documento podrá ser público para toda la compañía. Por ejemplo, los Documentos de Compliance siempre serán públicos, aunque NO todos los empleados deban indicar el “Leído y comprendido”.
- Una vez revisado y aprobado un Documento, la publicación del mismo (si así se ha configurado para dicho Documento) provocará el lanzamiento de una distribución electrónica, para la aceptación de

contenido de “Leído y comprendido” por parte de los Usuarios que se hayan configurado a nivel del Documento.

- El Sistema lanzará un acceso (link) de aceptación de requisitos, pero no tendrá modo de obligar al usuario receptor a abrir dicho acceso.
- El Documento deberá ser de lectura para todos los usuarios, impidiendo su modificación.

### **1.3.2 Documentos Internos**

Documentos que deben almacenarse para que puedan ser consultados por parte de todos los Usuarios de la compañía, sin restricción.

- Tal y como se ha indicado, todos los Documentos de este tipo serán sin restricción para el personal de la compañía. No obstante, en el caso de que un Documento sea claramente confidencial, se podrá restringir el acceso por grupos de Usuario (acceso con o sin restricción).
- Adicionalmente, se dejará previsto el que dicha Documentación pueda ser restringida a nivel de grupos de Usuarios. Esto obedece, más que a un tema de Seguridad, a una cuestión de agilidad, de forma que un Usuario sólo visualice los Documentos que le aplican, agilizando así, por ejemplo, la tarea de localización de un Documento concreto. Para este caso, se podrá configurar determinados colectivos que tendrán la potestad de consultar todos los Documentos (Compliance, Asesoría Jurídica, etc.).
- Este tipo de Documento no precisa la funcionalidad de “Leído y comprendido”.
- El Documento deberá ser de lectura para todos los usuarios, impidiendo su modificación.

Para los documentos nombrados anteriormente se han definido dos tipos de bibliotecas, una para documentos controlados y otra para documentos internos, teniendo cada biblioteca las mismas propiedades.

## **1.4. Solución ofrecida**

Una vez analizados los requerimientos básicos que se precisan, se determina que el gestor documental se realizará utilizando la plataforma de SharePoint.

## **CAPÍTULO 2. Plataforma de la solución**

En este capítulo se presenta la plataforma sobre la cual se desarrollará la solución. Se hará una breve introducción sobre la plataforma y se mostrará la evolución que ha seguido mediante su historia.

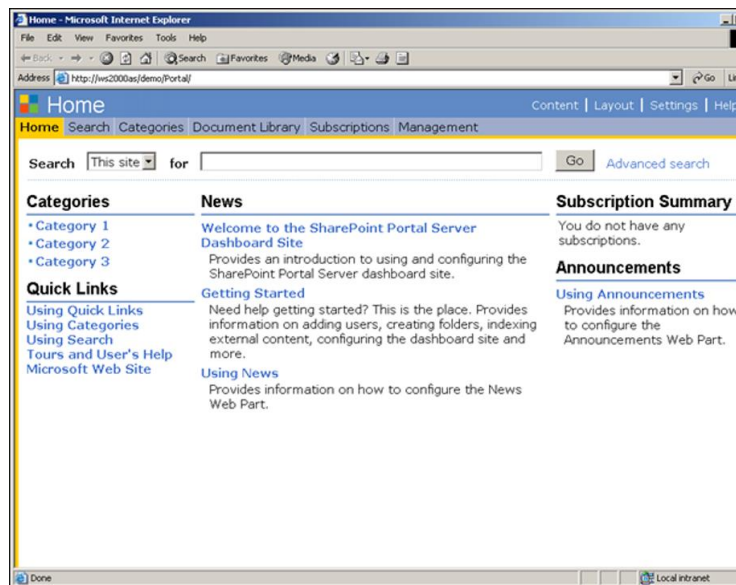
### **2.1 Sharepoint**

La solución se desarrolla sobre la plataforma de Microsoft diseñada para la implementación de portales de documentación. Esta plataforma se conoce como SharePoint y es un producto que, además de esta funcionalidad, tiene otras como la de colaboración empresarial con funciones de colaboración entre empleados, módulos de búsqueda y elementos de comunicación. Contiene un gran conjunto de herramientas de Microsoft que permiten, de forma ágil, flexible y escalable, construir soluciones de intranet, portal del empleado, portales web con una potente gestión de contenidos. Esta herramienta abarca un sinnúmero de soluciones empresariales que tengan relación con la comunicación, gestión del conocimiento y la inteligencia del negocio.

### **2.2 Historia**

#### **2.2.1 SharePoint Portal Server 2001**

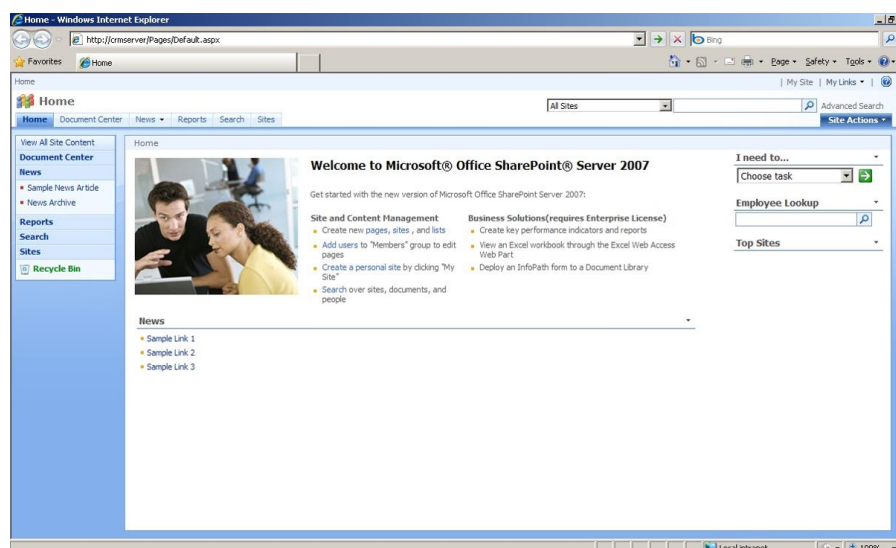
Esta herramienta nació en 2001 para ofrecer a los usuarios una herramienta versátil para la colaboración y productividad. Empezó a utilizarse para almacenar documentos y organizar archivos, especialmente orientada a documentos Microsoft Office.



**Ilustración 2.1. Pantalla SharePoint Portal Server 2001**

## 2.2.2 SharePoint Portal Server 2003

Esta nueva versión proveía de mayores herramientas de colaboración, una mejor interfaz, funciones de búsqueda y la habilidad de poder personalizar el producto, haciéndolo un portal escalable permitiendo utilizar las mismas herramientas de desarrollo que otros productos Microsoft.



**Ilustración 2.2 Pantalla SharePoint Portal Server 2003**

### 2.2.3 Office SharePoint Server 2007

Esta es la primera versión que alcanzaría el éxito tras poder ofrecerlo a través de CD's con licencias de hasta 25 usuarios de forma gratuita. Además, permitía la instalación en una única máquina aparte de otras mejoras de administración de contenidos.

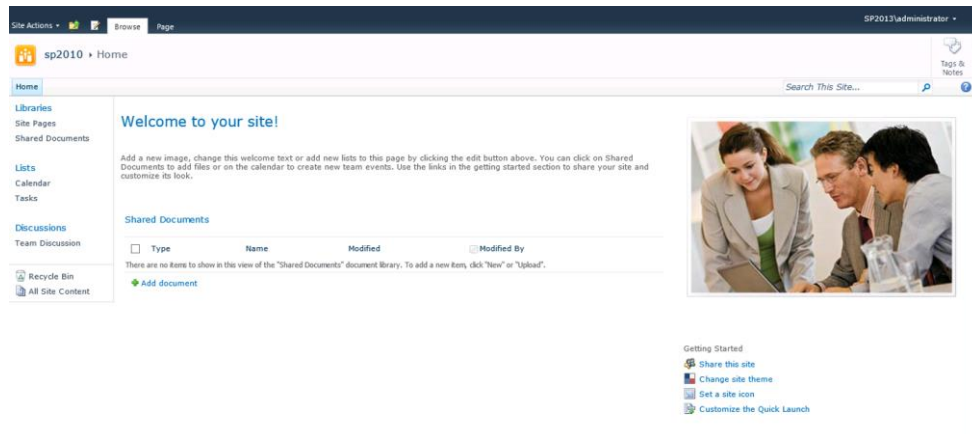


Ilustración 2.3. Pantalla Office SharePoint Server 2007

### 2.2.4 SharePoint Server 2010

Esta versión implicó una mayor integración entre todos los productos de la suite de Microsoft. Además de ello, incluyó mejoras notables como nuevas opciones para Workflows, wikis, perfiles sociales y características de Networking, aparte de mejoras en la interfaz y mayor soporte a un amplio número de navegadores.



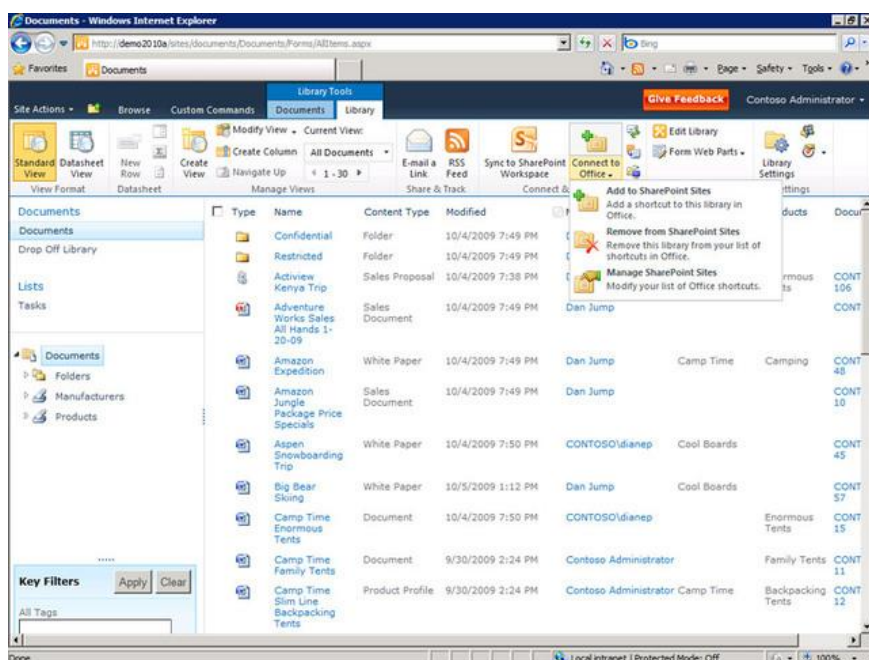


Ilustración 2.4. Pantalla Office SharePoint Server 2010

## 2.2.5 SharePoint Server 2013

Una de las versiones más robustas en la actualidad debido a las mejoras que se presentaron que hicieron que fuera un cambio incremental más que una nueva versión. Se realizaron pequeñas mejoras, arreglo de bugs y mejoras en el cacheo de servicios aparte de una nueva UI.

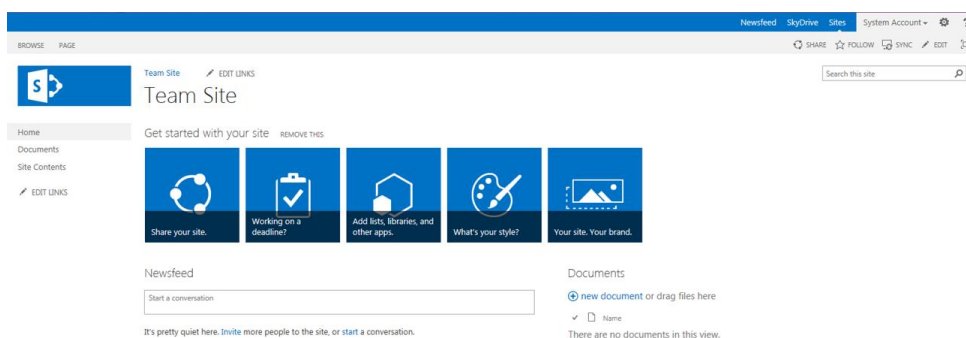


Ilustración 2.5. Pantalla Office SharePoint Server 2013

## 2.2.6 SharePoint Server 2016

La última versión disponible por Microsoft orientada más a la nube donde la permisividad para la creación de entornos híbridos junto a Office 365 la hacen destacar en este entorno.

Pasa a ser del referente de Microsoft a recibir las actualizaciones tardías en boga a su propio servicio de Office 365.

## 2.2.7 SharePoint Online – Office 365

Microsoft crea un grupo de software más suscripciones a servicios, donde a los consumidores se les permitía el uso de las aplicaciones de Microsoft Office, espacio de almacenamiento en OneDrive, 60 minutos gratuitos de Skype al mes. Además, a las empresas se les podía proveer de servicios sociales y de emailing con las herramientas Exchange, Skype for Business, SharePoint y, posteriormente, Yammer. En esta versión no encontramos versiones ya que Microsoft va actualizando periódicamente que van actualizando SharePoint.

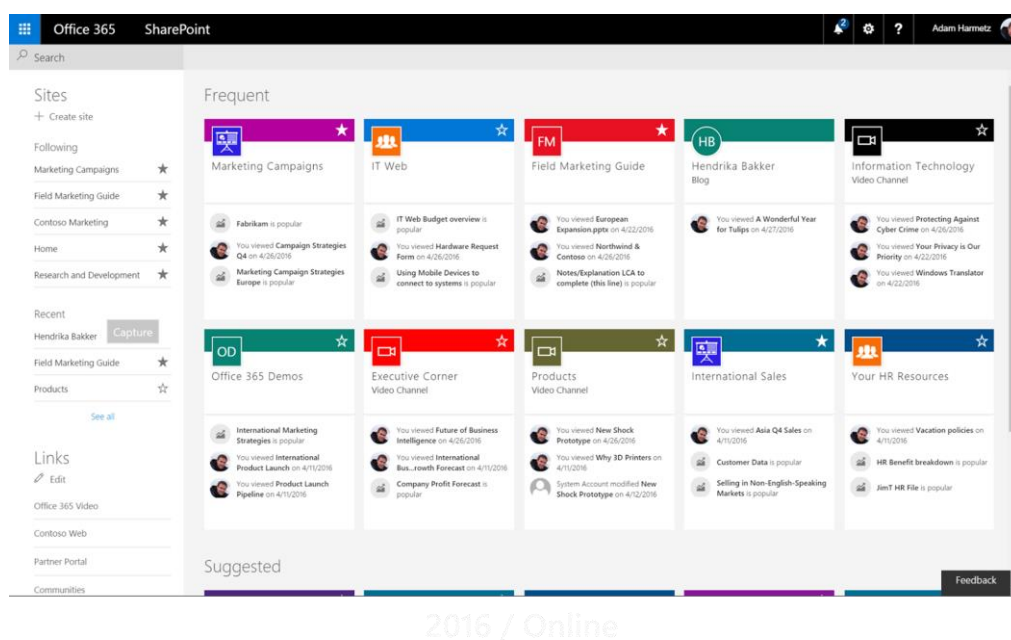


Ilustración 2.6. Pantalla SharePoint Online

## **CAPÍTULO 3. Arquitectura**

En este capítulo se define la arquitectura que se desarrollará en la solución. Para ello se detallará cual es modelo de datos que compone la solución. Por último, también se hablará sobre la seguridad y permisos que ofrece.

La arquitectura de la solución se basa en los estándares de diseño y programación definidos por Microsoft para el desarrollo y publicación de portales documentales.

### **3.1. Modelo de objetos**

SharePoint ofrece un modelo de objetos del lado servidor altamente estructurado. A continuación, se detallarán los diferentes tipos de objetos que podemos encontrar acompañados de una breve descripción.

#### **3.1.1 Granja**

Una granja es un conjunto de servidores que trabajan de forma integrada para proveer de servicios básicos de SharePoint. Algunos de los principales tipos de servidores que se pueden encontrar son:

- Servidor Web
- Servidor SQL
- Servidor Aplicaciones
- Servidor Correo

#### **3.1.2 Base de datos de contenidos**

Base de datos que contiene los datos de los usuarios para una aplicación web concreta.

#### **3.1.3 Aplicación Web**

Web almacenada en un servidor IIS. Aloja colecciones de sitios y provee de los accesos mediante credenciales y otras configuraciones a lo largo de las aplicaciones que existan en la granja.

#### **3.1.4 Servicios**

Aplicación que está ejecutándose en segundo plano en una granja de servidores.

### 3.1.5 Colección de sitios

Conjunto de sitios existentes en una aplicación web y, a su vez ésta, se desglosa en sitios y subsitios de forma jerárquica.

### 3.1.6 Sitio

Web que administra contenidos en su interior: listas, bibliotecas, webparts, etc...

### 3.1.7 Lista/Biblioteca

Objetos que permiten almacenar información (y documentos) en su interior como elementos de lista.

Entorno – Productivo

Roles y Hardware(por servidor)	Número de Servidores	Servicios por Servidor	Notas
<b>Front end</b> VM, 4 cores, 16 GB RAM HDD-SQ: 120GB HDD-Paginación: 25GB HDD-Datalog: 30GB		Microsoft SharePoint Foundation Web Application Request Management Search Query	Dos servidores para Alta Disponibilidad. Se requiere Balanceador de Carga para balancear las peticiones entre los servidores.
<b>Application + Search Crawl</b> VM, 4 cores, 32 GB RAM HDD-SQ: 120 GB HDD-Paginación: 50GB HDD-Logindex: 100GB		Crawl Target Microsoft SharePoint Foundation Web Application Machine Translation Central Administration PowerPoint Conversion User Profile Synchronization Word Automation Work Management Workflow timer service Access Services and Access Services 2010 Business Data Connectivity	Servidor de indexación, alta utilización de recursos, no requiere una latencia muy alta. No procesa peticiones de usuario. Alta utilización de recursos. No procesa peticiones de usuario.
<b>Database</b> 4 cores, 16 GB a instancia HDD-SQ: 80GB HDD-Paginación: 30 GB HDD-BBDD-Data: 100 GB HDD-BBDD-Log: 50 GB HDD-BBDD-TempDB: 30 GB HDD-BBDD-Backup: 100 GB		SQL Server – BBDD	SQL Server donde reside toda la configuración y contenido de la plataforma SharePoint Server <b>Instancia dedicada</b>

<b>Cuentas de Servicio necesarias (nomenclatura ejemplo entorno -producción):</b> - spAdmin - spFarm - spDistCache - spService - spWeb - spExcel - spExcelSvc - spPivot - spSandbox - spSearchSvc - spSearchCA - spSuperReader - spSuperUser - spUnattSvc - spUpsync - spClaims	<b>Cta. Instalación de la granja (Admin local de todos los servers)</b> Cta. Administradora de la granja Cta. Cache Distribuida Cta. General Application Pool Cta. General Web Application Pool Cta. Excel Services Application Pool Cta. Excel Services Cta. PowerPivot Cta. Sandbox Código de espacio aislado. Cta. Servicio de Búsqueda Cta. Contenido de Búsqueda Cta. SuperReader SharePoint Cta. SuperUser SharePoint Cta. Servicio Desatendido Cta. User Profile Services (Necesita permisos AD) Cta. Servicio Claims to Windows Token Service.	<b>Software Necesario:</b> - Windows Server 2012 R2 instalado y actualizado. - SQL Server 2014 (Recomendado inglés) - SharePoint Server 2013 Standard Edition with SP1 (Recomendado en Inglés) - Language Pack SPS2013 SP1 (Descarga disponible) - Cumulative Updates Mayo 2015 mínimo (Descarga Disponible)  <b>Licenciamiento entorno</b> - 1 Licencia SQL Server Standard 2014 (por core) – Adjuntamos PDF con el licenciamiento de SQL Server 2014 - 2 Licencias SharePoint Server 2013 SP1 - 3 Licencias Windows Server 2012 R2  <b>Otros requerimientos:</b> - Servidores en dominio - Conectividad a Internet de los servidores en el momento de la instalación - Certificados SSL para la configuración de la Aplicación Web de SharePoint. - Acceso a Directorio Activo (o recurso de IT con permisos) para configurar Kerberos
---	---	--

Ilustración 3.1 Esquema de servidores

### 3.1.8 Modelo de datos

Un modelo de datos describe piezas de información del mundo real con el que trabajarán nuestras aplicaciones a través de una relación entre las diferentes piezas. En SharePoint existen varias entidades básicas.

### 3.1.9 Listas

Son el mecanismo principal de almacenamiento en la plataforma. Están compuestas por columnas y filas (elementos), y permite crear relaciones entre listas.

### 3.1.10 Columnas

Son el núcleo de construcción en SharePoint. Permite que se le apliquen propiedades básicas (nombre y tipo), de agrupación, lógicas (requerido, único, longitud máxima del campo, valor por defecto) e, incluso, reglas de validación. Pueden ser de diferentes tipos: string, int, Campo calculado a partir de otras columnas, etc.

### 3.1.11 Tipos de contenido

Son una agrupación lógica de una o más columnas. Sirven para añadirles flujos de aprobación, políticas de retención de información y, sobre todo para compartir metadatos comunes entre diversos elementos de una única colección de sitios.

Para poder tener una mejor aproximación de estos elementos, podemos hacer una comparación con elementos que se encuentran en SQL.

SharePoint	SQL
Listas	Tablas
Columnas	Columnas
Tipos de contenido	Conjunto de columnas
Elementos	Filas

### 3.1.12 Seguridad y permisos

La seguridad es un añadido que ofrece SharePoint de base con respecto a una simple aplicación .NET. Está basada en grupos o usuarios, y roles de permisos que se pueden crear y combinar de múltiples formas. SharePoint, por defecto, genera grupos de Visitantes (lectura), Miembros (edición) y Propietarios (administración) para cada uno de los sitios que se crean. Los permisos se aplican jerárquicamente de arriba a abajo y permiten controlar las acciones de los usuarios sobre los elementos. Adicionalmente, esta jerarquía de permisos se puede romper para cualquier objeto y aplicar permisos específicos para ellos.

## **CAPÍTULO 4. Desarrollo de la solución**

En esta segunda parte se procederá a explicar cómo se ha desarrollado la solución. Como se ha explicado anteriormente la solución se desarrolla para un entorno de SharePoint, en este caso la compañía para la cual se desarrolla dispone de un SharePoint On-Premise 2016. Primero se definirán las herramientas que se han utilizado en el desarrollo y, posteriormente se hablará sobre el diseño de la solución. Para concluir se detallará el proceso de creación de la solución, es decir, como se ha desarrollado la estructura y la lógica de la solución.

### **4.1 Herramientas utilizadas**

Para el desarrollo de la solución en SharePoint se ha utilizado la herramienta de desarrollo Visual Studio, aparte del propio entorno de SharePoint que el cliente ya tiene. Sobre SharePoint se ha comentado en apartados anteriores sobre esta herramienta.

#### **4.1.1 Visual Studio**

Es un entorno de desarrollo integrado, el cual soporta múltiples lenguajes de programación y entornos de desarrollo web. En este proyecto en particular se hace uso de la versión 2017.

### **4.2 Diseño de la solución**

#### **4.2.1 Funcionalidades de la solución**

Las Funcionalidades de la solución son las características detalladas del producto final que permiten conocer qué acciones ofrece el entorno implementado, así como sus posibilidades de parametrización y configuración. El conjunto de estas funcionalidades conforma el alcance final del proyecto acordado.

- FS01: Carga y consulta de documentos
- FS02: Funcionamiento de listas de estado
- FS03: Formulario de leído y comprendido
- FS04: Restringir acceso a documentos
- FS05: Proceso para nuevas incorporaciones
- FS06: Proceso de recordatorios

## 4.2.2 Diseño conceptual

El Diseño Conceptual establece los elementos de alto nivel de la solución implementada y permiten tener una primera visión comprensible para todo el equipo de qué forma se va a crear la solución. En la ilustración 4.1 podemos ver el proceso que sigue la subida de un documento.

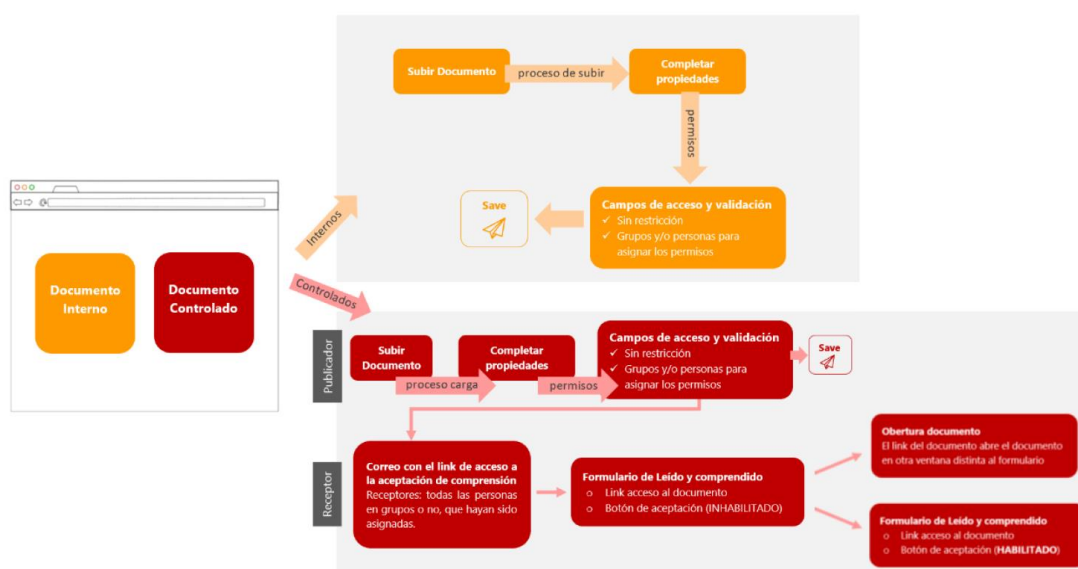


Ilustración 4.1. Proceso publicación de documentos

### 4.2.2.1 Carga y propiedades del documento

La carga de documentos se realiza mediante el botón de “Nuevo documento” y no se puede realizar masivamente. Esta limitación es para permitir realizar el completado de propiedades correctamente, y por ello, se deben que cargar y completar las propiedades por cada archivo.

En caso de no completar todas las propiedades obligatorias, el documento quedará almacenado en la biblioteca, pero sólo será visible para el usuario que ha publicado y quedará desprotegido hasta completar dichas propiedades. Una vez completadas, el propietario deberá hacer “Check-in” del documento para cambiar el estado a protegido y hacerlo visible para el grupo de personas indicado en las propiedades de este. Actúa de igual manera, en caso de que se cierre el formulario sin haber completado todas las propiedades (también quedando desprotegido, es decir, no visible públicamente para el resto de los usuarios).

Si a futuro, se añaden campos de propiedades (columnas a la lista) en la biblioteca, no habrá histórico para dicho campo, sino que se empezará a almacenar desde el momento de la creación del campo.

El campo de “Nombre del Documento”, por estándar de SharePoint, no es modificable, por tanto, si se desea modificar el nombre se debe eliminar el erróneo y volver a cargar el archivo correctamente.

Si un mismo documento aplica a dos idiomas, se han de publicar dos documentos para generar dos elementos. Ya que la propiedad de “Idioma” aplica al nombre del documento y éste es único.

Los Tags que categorizan un documento, se pueden añadir al subir el mismo en el gestor documental, o bien se pueden editar posteriormente en las propiedades de este. Los Tags se configuran como metadatos, por la/las personas que tengan permisos especiales para hacerlo, en el menú cerrado correspondiente de SharePoint (Term Store). Esto significa que no se puede añadir un Tag a un documento que no exista como metadato. Por lo que primero habrá que añadir los Tags deseados como metadatos y posteriormente introducirlos en las propiedades de los documentos. Esto es así para evitar que distintos usuarios den un valor distinto a un mismo concepto, por temas idiomáticos, mayúsculas o errores ortográficos.

Las personas que pueden añadir Tags (existentes como metadato) a documentos son las personas con permisos para publicar documentos.

Una vez añadidos los tags deseados a los detalles de cualquier documento, estos no se mostrarán en la lista de documentos, para verlos, habrá que entrar en las propiedades del documento.

En las propiedades de un documento, a petición de la compañía, no existe un campo de “agrupación”, y, por tanto, no se puede filtrar documentos por un concepto de este tipo.

Para los documentos controlados, una vez el usuario recibe el correo que genera el sistema para la validación de los documentos (“Leído y comprendido”), este es redirigido al formulario de “leído y comprendido”. En este formulario se encuentra un enlace al documento y un botón deshabilitado para la aceptación de las condiciones. Dicho botón únicamente se habilita cuando el usuario abre el enlace que del documento que tiene que leer y comprender.

Si el usuario rechaza el formulario de aceptación y no responde, queda como “No leído y comprendido”.

#### **4.2.2.2 Listas relacionadas y comportamiento con las propiedades**

Se han establecido tres listas de selección:



- Lista de departamento
- Lista de tipo de documento
- Lista de idiomas

El acceso a estas listas es restringido, de manera que, sólo pueden acceder las personas que tengan permisos para editarlas.

Cada una de estas listas serán definidas por la compañía y corresponderán a una opción del campo seleccionable en la lista principal de Documentos. Adicionalmente, las listas nombradas anteriormente contienen otra columna con la abreviatura de la propiedad, definida por la compañía, que es utilizada para la generación del “Código Alfanumérico”

El Código alfanumérico se utiliza para identificar de manera unívoca cada uno de los documentos almacenados en el Gestor Documental. Dicho código estará compuesto por:

- Año de publicación (dos últimas cifras)
- Abreviatura de tipo documento (de la lista Tipo de Documento)
- Abreviatura del departamento (de la lista Departamento)
- Abreviatura idioma (de la lista de Idiomas)
- Código enumerativo creado automáticamente (incremental de 1)

El Código enumerativo se incrementa respecto al código generado anteriormente en la misma biblioteca, independientemente de las demás propiedades, de manera que cada biblioteca puede tener el mismo código enumerativo.

El código alfanumérico tiene el siguiente formato:

Ejemplo → 18-PO-COM-ES-1

Para que el nombre del documento sea unívoco por cada documento, el nombre final será la unión del Código alfanumérico y el nombre original con el que se ha subido el documento:

Ejemplo → 19-PO-COM-ES-4\_Política de riesgos.pdf

Si un documento tiene una nueva versión, se genera un nuevo código para dicho documento (distinto al que tiene el documento anterior). De tal manera que, para informar a los usuarios que existe una versión de un documento, el promotor del mismo tendrá que activar la propiedad de Vigente No para las versiones anteriores del propio documento y el sistema publica el nuevo documento con esta propiedad activada.

#### **4.2.2.3 Ciclo de vida del documento**

En caso de que sea un Documento Controlado (siempre que se genere un nuevo documento), se procede a lanzar el proceso de Nuevo Documento,

enviando un link a los usuarios indicados para que acepten el “Leído y comprendido”.

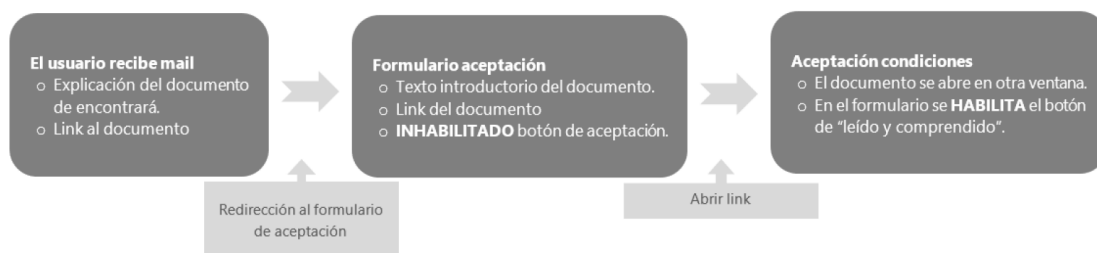
Para saber las personas que han aceptado el contenido del documento, o para saber la información relacionada con el estado, se muestra un listado que almacena dicha información (Lista de Estados de Documentos).

Si se desea asignar un documento a un grupo de personas que no está creado, las acciones a realizar son:

- Ponerse en contacto con el Departamento de IT para que lo creen.
- Subir el archivo (una vez el grupo ya está creado), así se puede asignar el nuevo grupo.



**Ilustración 4.2 Flujo de funcionamiento para la lista de estado**



**Ilustración 4.3 Flujo formulario leído y comprendido**

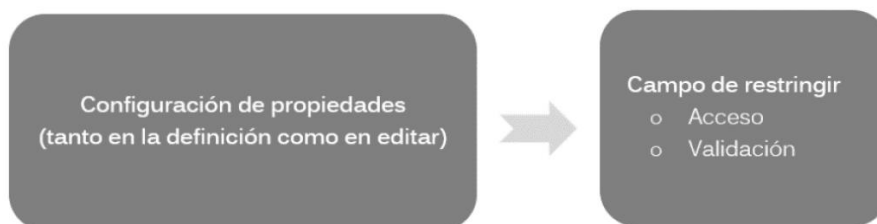


Ilustración 4.4 Flujo funcionamiento restringir documentos

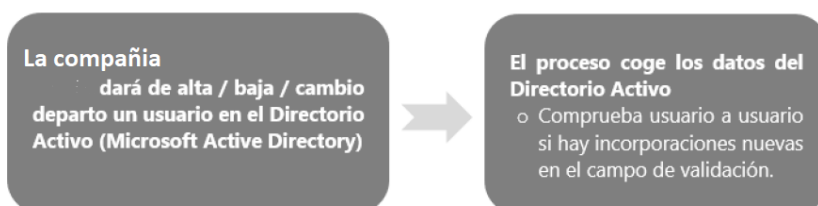


Ilustración 4.5 Proceso nuevas incorporaciones

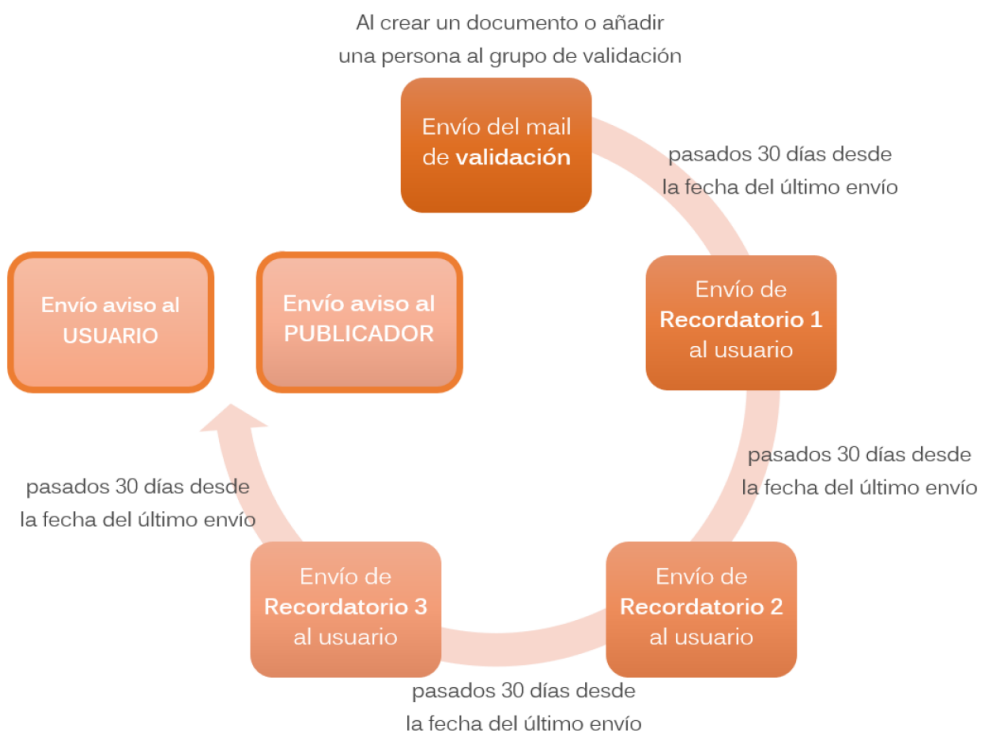


Ilustración 4.6 Proceso de recordatorios

Para indicar el número de días que deben transcurrir para enviar los recordatorios, se puede modificar en una Lista de Configuración. Esta lista aplica los cambios a todos los recordatorios, es decir, se indican los días que transcurren respecto la última notificación.

También son configurables los cuerpos de los mensajes que se envían tanto a los usuarios que no hayan complementado debidamente el indicador de “Leído y Aceptado” como a los promotores del documento, en caso de ser necesario.

## **4.3 Diseño físico**

El Diseño Físico define los elementos finales de la solución, de forma que el equipo de Desarrollo tenga establecidos los elementos que deben ser implementados en las tareas asignadas

### **4.3.1 Fases de la solución**

En la solución implementada se pueden distinguir tres fases diferentes, desarrolladas por las herramientas que se han comentado anteriormente. Estas fases son:

- La solución se desarrolla en un nuevo site del SharePoint, por lo que, para el desarrollo de la solución, se crea uno nuevo que alberga la solución ofrecida, donde se publican los documentos para que, posteriormente, puedan ser consultados por los usuarios de la compañía.
- Una vez creado el site, se instala una solución, desarrollada con Visual Studio, que despliega la estructura y la lógica del proyecto.
- Por último, una vez desplegada la solución, se realizan los cambios para acabar de ajustar la solución mediante la incorporación de archivos JavaScript en algunos elementos de la solución.

## **4.4 Creación del site de SharePoint**

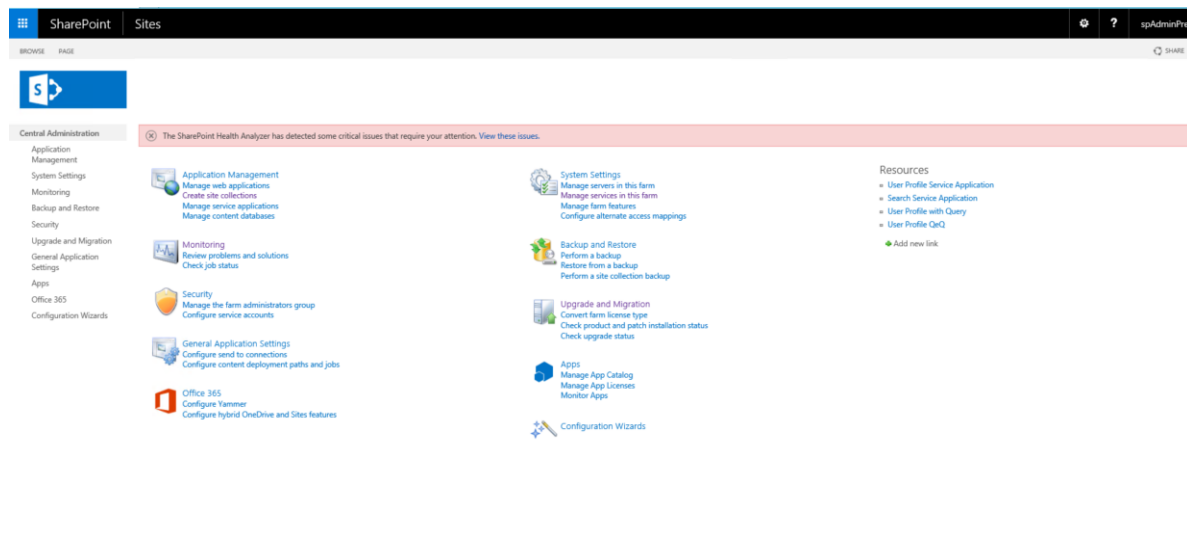
En este primer punto se procede a la creación de un nuevo site que alberga la solución. Para poder crear un nuevo site, primero de todo es necesario que el usuario con el que se quiera crear el site tenga los permisos adecuados, en este caso el usuario debería tener permisos de administrador del site.

En el caso concreto de un SharePoint On-Premise, para poder administrar la creación de un nuevo site tiene que ser creado a través de la Administración Central de la plataforma.

A través de ella los usuarios pueden realizar diferentes tareas de administración como:

- Gestión de seguridad
- Configuraciones del sistema
- Backup
- Etc...

Para poder acceder a ella es necesario estar conectado en un ordenador desde donde se tenga acceso a la configuración del sistema, ya que podremos ejecutar la central admin, dado que se encuentra instalada como un programa. Por lo que simplemente hay que buscar en el ordenador y ejecutar el programa. Una vez cargue, aparecerá una pantalla con diferentes opciones.



**Ilustración 4.7. Pantalla central admin**

En la pantalla hay que rellenar información sobre el nuevo site que se creará. A continuación, se mostrarán los datos que son requeridos para la creación del site, junto con la información que se utilizó para crear el site de este proyecto.

- **Título:** Título que tendrá el nuevo site. En el proyecto se eligió el nombre de Gestor Documental
- **Descripción:** Información acerca del nuevo site.
- **URL:** La url desde donde se accederá al site. La primera parte viene predefinida al instalar el SharePoint. En el proyecto se dio la url relativa /sites/gestordocumental

- **Lenguaje:** El lenguaje en el que se instala el site. Toda la configuración que se haga en el site aparece en el idioma elegido. En el proyecto se configuró el idioma en español.
- **Plantilla:** SharePoint nos da a elegir diferentes tipos de plantillas, según el tipo de plantilla modificará las características y diseño del sitio. Para el tipo de proyecto que se quería crear la plantilla que elegimos fue la de Team Site.
- **Administrador:** En esta parte nos da para elegir un usuario de SharePoint que es el administrador de ese site. Como administrador se puede configurar las características del site y tiene un control total sobre dicho espacio. En el proyecto se configuró un usuario que nos dio la compañía para que tuviéramos acceso al SharePoint y pudiéramos configurar este nuevo site.

**Ilustración 4.8. Pantalla creación de site**

#### 4.4.1 Programación de la solución

Una vez creado el espacio, se procede al desarrollo de la solución. Para el desarrollo de la situación se ha utilizado Visual Studio. Tal y como se detalla en el apartado anterior la solución consta de dos bibliotecas, además de unas listas de configuración.

Para el caso del proceso de caducidad de documentos se utiliza un TimerJob, y para el caso de envío de recordatorios se utiliza un Worker. Por último, para el caso de cambio de nombre del documento que se suba se hará uso de un Receptor de Eventos.

La solución está formada por dos paquetes, uno de la estructura del proyecto y otra de los Timer Jobs, que se instalan en SharePoint mediante diferentes Features. Cada una de estas Features despliega una parte diferente del proyecto.

A continuación, se detallan las diferentes Features que componen los dos paquetes de la solución.

## **4.5 Creación de la solución**

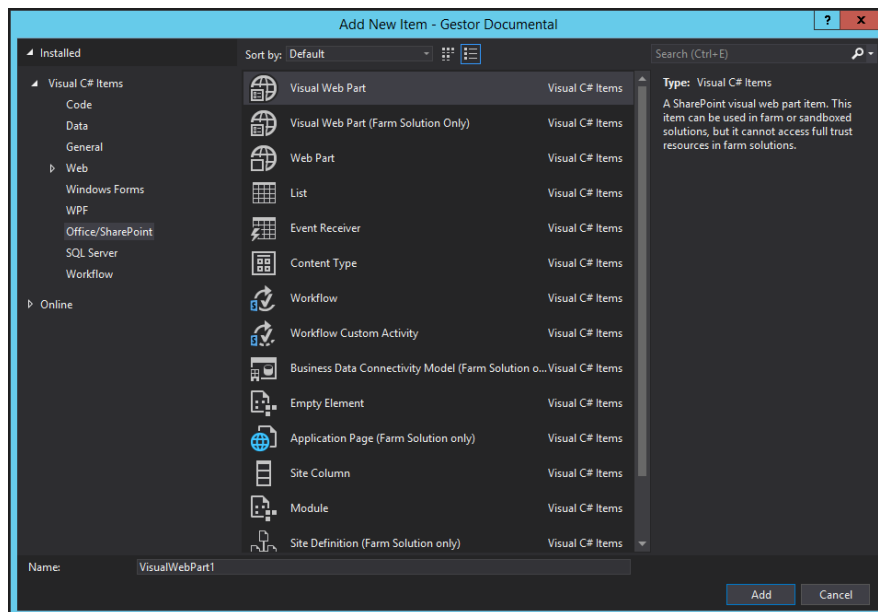
Para la creación de la solución hay que entrar en el Visual Studio, seleccionar Nuevo proyecto y solución vacía de SharePoint 2016, la versión dependerá de la versión de SharePoint donde se tenga que desplegar [1].

Una vez fue creada la solución se crearon dos proyectos diferentes, uno con toda la parte de estructura, y otro con los Timer Jobs. Esto permite poder desplegar de manera independiente la estructura de los TimerJobs, un punto importante a la hora de actualizar la solución, ya que de esta manera los desarrolladores pueden actualizar partes de la solución sin necesidad de tener que hacerlo de todo el proyecto.

Por último, también se creó un proyecto de tipo de biblioteca de clase donde se encuentran funciones comunes que pueden ser utilizadas en las dos soluciones.

### **4.5.1 Columnas de sitio**

En esta Feature se han programado todas las columnas de sitio que luego han sido utilizadas en las listas o en las bibliotecas. Para programar las diferentes columnas hay que hacer clic en la solución donde se creara la estructura y hay que seleccionar la opción de añadir un nuevo ítem, que al seleccionarla mostrara un menú con todos los elementos que podemos crear. En este caso se seleccionó la opción de Site Column.



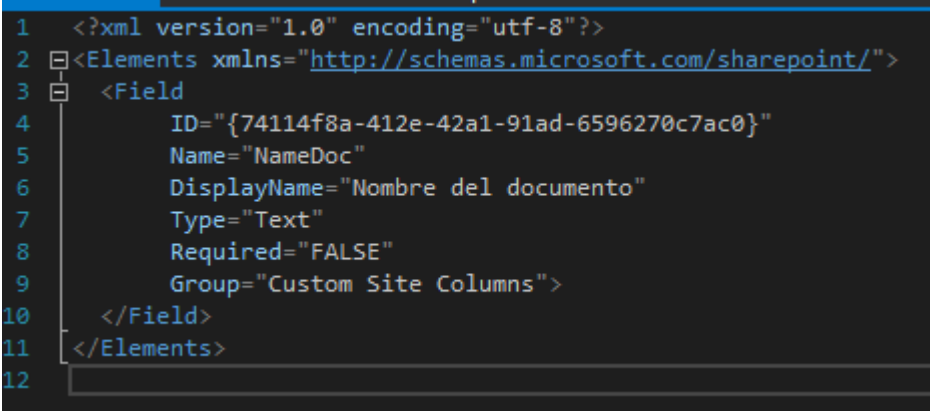
**Ilustración 4.9** Formulario creación nuevo ítem

Una vez creado el objeto en la solución, al entrar en el objeto, se puede ver una estructura de xml donde hay diferentes propiedades del objeto. Estas propiedades son:

- *ID*: Es un guid que identificara al objeto de manera única e inequívoca. Esta propiedad no puede ser cambiada, y es asignada de manera aleatoria.
- *Name*: Nombre interno del objeto, también nos permitirá identificar el objeto a través de él. Esta propiedad puede ser modificada antes de crear el objeto, una vez creado no puede volverse a modificar.
- *DisplayName*: Es el nombre que SharePoint utilizará para mostrar el objeto. Puede ser diferente del Name y también puede ser modificado antes y después de ser creado.
- *Type*: En esta propiedad definimos el tipo de columna que será. Aquí podemos encontrar diferentes tipos como de usuario, texto, numero, ... Una vez creada la columna no se puede volver a cambiar el tipo.
- *Required*: Nos permite definir si esta columna será requerida, es decir, cuando se cree un ítem que contenga esta columna nos obligará a introducir información de todas las columnas que tengan esta propiedad. Esta propiedad se marca como un booleano, si se marca como true la columna será de tipo requerido. Esta propiedad también puede ser cambiada después de ser creada.
- *Group*: Esta es la última propiedad permite agrupar columnas que tengan un mismo grupo, es decir, todas las columnas que tengan un



mismo grupo, SharePoint las agrupará, de tal manera que a la hora de crear una lista o biblioteca podemos indicar a SharePoint que nos muestre todas aquellas que pertenezcan a un grupo en concreto, de tal manera que sea más sencillo a la hora de crear una lista o biblioteca.



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <Elements xmlns="http://schemas.microsoft.com/sharepoint/">
3   <Field
4     ID="{74114f8a-412e-42a1-91ad-6596270c7ac0}"
5     Name="NameDoc"
6     DisplayName="Nombre del documento"
7     Type="Text"
8     Required="FALSE"
9     Group="Custom Site Columns">
10   </Field>
11 </Elements>
12
```

Ilustración 4.10. Estructura del elemento en formato xml

#### 4.5.2 Contenidos del sitio

Una vez que se crearon las columnas de sitio se pudo proceder a crear los contenidos del sitio. Para poder crear contenidos del sitio se tuvo que volver a añadir un nuevo ítem, en este caso se seleccionó la opción de Content Type. En este caso aparecerá una tabla con tres columnas.

- **DisplayName:** Al hacer clic sobre ella nos permite seleccionar una columna que haya sido creada con anterioridad o una columna de las que SharePoint tiene ya predeterminadas.
- **Type:** En esta columna se muestra de que tipo es la columna que hemos seleccionado. Se completa de manera automática una vez elegida la columna.
- **Required:** Nos aparece una casilla que podemos marcar. En el caso que marquemos la casilla hará que esa columna sea requerida en el contenido de sitio que se cree.


Display Name	Type	Required
Departamento	Lookup (information already or	<input type="checkbox"/>
Tipo de Documento	Lookup (information already or	<input type="checkbox"/>
Idioma	Lookup (information already or	<input type="checkbox"/>
Tags	Single Line of Text	<input type="checkbox"/>
Fecha de publicacion	Date and Time	<input type="checkbox"/>
Vigente	Boolean	<input type="checkbox"/>
Detalles	Single Line of Text	<input type="checkbox"/>
Promotor	Multiple Persons or Groups	<input type="checkbox"/>
Fecha de caducidad	Date and Time	<input type="checkbox"/>
Personas leído y comprendido	Multiple Persons or Groups	<input type="checkbox"/>
Codigo	Single Line of Text	<input type="checkbox"/>
Personas con acceso publicacion	Multiple Persons or Groups	<input type="checkbox"/>
Empresas	Multiple Value Lookup	<input type="checkbox"/>
Nombre del documento	Single Line of Text	<input type="checkbox"/>
 Click here to add a column		<input type="checkbox"/>

Ilustración 4.11. Tabla de Contenido de Sitio

### 4.5.3 Listas y bibliotecas

El siguiente paso que se siguió fue la creación de las listas y las bibliotecas. Al igual que en los casos anteriores se creó un nuevo elemento de tipo List. En este caso aparece otra pantalla que permitirá seleccionar que tipo lista se quiere crear. Para el caso de la lista se seleccionó la opción que viene por defecto, que es una Custom List. Para el caso de la biblioteca se debe seleccionar Document Library. Una vez creados los elementos, se le añadieron los contenidos del sitio, que se crearon previamente, para cada lista o biblioteca. Para añadir los contenidos hay que seleccionar el botón de Content Types y se puede añadir un contenido del sitio que previamente haya sido creado.

SharePoint Customization Wizard

Choose List Settings

What name do you want to display for your list?  
List1

Do you want to create a customizable list template or a list instance based on an existing list type?

☒ Create a customizable list template and a list instance of it:  
Document Library

☐ Create a list instance based on an existing list template:  
Documentos Controlados (Gestor Documental\ControlledDocs)

< Previous   Next >   Finish   Cancel

Ilustración 4.12. Formulario creación de lista

Use the grid to configure columns for the list. [Learn more about creating lists](#)

Column Display Name	Type	Required
Departamento	Lookup (information already o	<input type="checkbox"/>
Tipo de Documento	Lookup (information already o	<input type="checkbox"/>
Idioma	Lookup (information already o	<input type="checkbox"/>
Tags	Single Line of Text	<input type="checkbox"/>
Fecha de publicacion	Date and Time	<input type="checkbox"/>
Vigente	Boolean	<input type="checkbox"/>
Detalles	Single Line of Text	<input type="checkbox"/>
Promotor	Multiple Persons or Groups	<input type="checkbox"/>
Fecha de caducidad	Date and Time	<input type="checkbox"/>
Personas leído y comprendido	Multiple Persons or Groups	<input type="checkbox"/>
Código	Single Line of Text	<input type="checkbox"/>
Personas con acceso publicacion	Multiple Persons or Groups	<input type="checkbox"/>
Empresas	Multiple Value Lookup	<input type="checkbox"/>

Click Content Types to add columns from an existing content type to your list, or to change the default content type.

Content Types

Use this template as the default when a user creates a new document in this library.

Ilustración 4.13 Configuración columnas de la lista

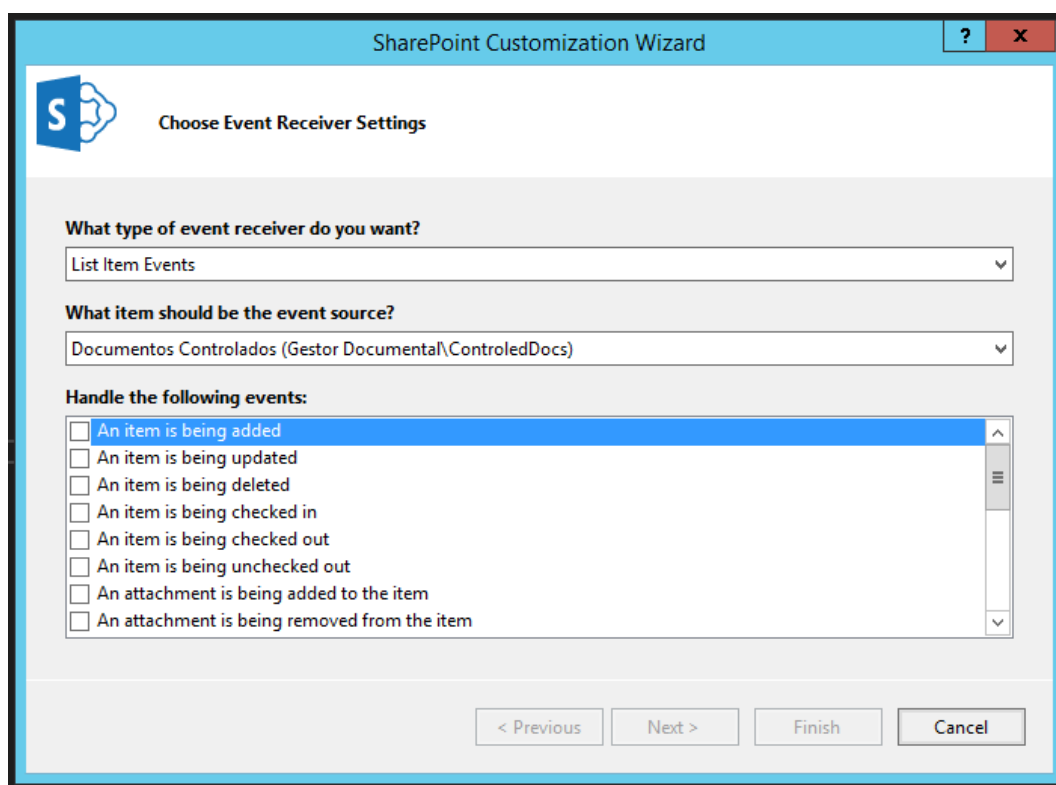
#### 4.5.4 Receptores de evento

Los receptores de eventos son una herramienta que nos ofrece SharePoint que permite poder ejecutar una lógica, programada por el usuario, cuando se realiza una acción sobre algún elemento de SharePoint. Un ejemplo de ello es ejecutar una lógica después que es un elemento se haya añadido o haya sido modificado de una lista en concreto. Para este proyecto se desarrollaron dos

receptores de eventos diferentes asociados a las dos bibliotecas que hay (Documentos controlados y Documentos Internos).

Para la creación de un receptor de eventos, al igual que en los casos anteriores hay que añadir un nuevo ítem de tipo Event Receiver. Una vez seleccionado aparecerá un Wizard donde hay que configurar tres parámetros:

- *El tipo de receptor de eventos:* Este parámetro hace referencia a que elemento se asocia el receptor. Los diferentes tipos que podemos encontrar son eventos de ítems de una lista, de lista, de web, de correo y de workflow. En este proyecto se utilizaron del tipo ítems de una lista.
- *Donde actuará el receptor de eventos:* En este parámetro se selecciona sobre que elemento actuará el receptor de eventos. Según el tipo de receptor de eventos podremos elegir el origen donde estará. Por ejemplo, si seleccionamos un receptor asociado a una lista, en este parámetro nos aparecerá una lista con todas las listas disponibles que tenemos en SharePoint. Solo podemos elegir un único parámetro, por lo que para cada elemento debemos crear un nuevo receptor.
- *Tipo de eventos asociados:* Que eventos hacen desencadenar la ejecución del receptor de eventos. El receptor de eventos solo actúa en unos eventos en particular (se elimina un ítem de una lista, se añade, se modifica, ...), este parámetro determina que eventos desencadenaran la ejecución del receptor. Este valor puede ser múltiple, de manera que podemos asociar diferentes acciones que desencadenen el receptor asociado a un elemento.



The screenshot shows the 'SharePoint Customization Wizard' window with the title 'Choose Event Receiver Settings'. It features the SharePoint logo and a list of settings to configure an event receiver. The first dropdown menu, 'What type of event receiver do you want?', is set to 'List Item Events'. The second dropdown menu, 'What item should be the event source?', is set to 'Documentos Controlados (Gestor Documental\ControlledDocs)'. Below these, a section titled 'Handle the following events:' contains a list of eight events, each with an unchecked checkbox. The first event, 'An item is being added', is highlighted with a blue background. At the bottom of the window are four buttons: '< Previous', 'Next >', 'Finish', and 'Cancel'.

SharePoint Customization Wizard

Choose Event Receiver Settings

What type of event receiver do you want?

List Item Events

What item should be the event source?

Documentos Controlados (Gestor Documental\ControlledDocs)

Handle the following events:

- ☐ An item is being added
- ☐ An item is being updated
- ☐ An item is being deleted
- ☐ An item is being checked in
- ☐ An item is being checked out
- ☐ An item is being unchecked out
- ☐ An attachment is being added to the item
- ☐ An attachment is being removed from the item

< Previous   Next >   Finish   Cancel

**Ilustración 4.14** Formulario creación receptores de eventos

Una vez se han seleccionado los parámetros ya se puede crear el nuevo receptor. Al crear un nuevo receptor aparecerá un nuevo ítem en la solución, si se entra a ver, se podrá comprobar cómo se ha creado uno o varios métodos. Estos métodos están asociados con las acciones que el usuario ha elegido al crear el receptor. Dentro del método se incluirá la lógica que se ejecutará cuando se realice la acción asociada, esta lógica se programa con el lenguaje .NET. A continuación, se procederá a explicar los dos receptores de eventos creados para el proyecto.

#### **4.5.5 Receptor documentos controlados**

Este receptor se ha programado para ejecutarse cuando haya un cambio en un ítem. La finalidad del receptor es de generar el código del documento a partir de las propiedades de este. Por otra parte, también rompe permisos de este ítem de manera que solo los usuarios que se han añadido como promotores, personas con acceso al documento, pendientes de leído y comprendido y los promotores. Por último, se pasa el ítem creado a uno de los Timer Jobs que se encargara de enviar los correos de leído y comprendido a los usuarios correspondientes.

Para el desarrollo de la lógica de este receptor se utilizó el Nuget de Microsoft.SharePoint.dll que fue necesario para realizar la lógica y poder trabajar con los diferentes elementos de SharePoint.

```

string name = li[SPBuiltInFieldId.LinkFilename].ToString();
string values = Path.GetFileNameWithoutExtension(name);
li[li.Fields.GetFieldByInternalName("DocumentName").Id] = values.Replace('.', '-');
DateTime localDate = DateTime.Now;
var codeYear = DateTime.Now.ToString("yy");
SPFieldLookup lookup = (SPFieldLookup)properties.List.Fields.GetFieldByInternalName("LanguageDoc");
Int32 singlelookupItemID = Convert.ToInt32(properties.AfterProperties["LanguageDoc"]);
string codeLang = properties.Web.Lists[new Guid(lookup.LookupList)].GetItemById(singlelookupItemID)["CodeConfiguration"].ToString();

SPFieldLookup lookupDoc = (SPFieldLookup)properties.List.Fields.GetFieldByInternalName("DocType");
Int32 itemIDDep = Convert.ToInt32(properties.AfterProperties["DocType"]);
string codeDep = properties.Web.Lists[new Guid(lookupDoc.LookupList)].GetItemById(itemIDDep)["CodeConfiguration"].ToString();

SPFieldLookup lookupDep = (SPFieldLookup)properties.List.Fields.GetFieldByInternalName("DepartmentType");
Int32 itemIDDoc = Convert.ToInt32(properties.AfterProperties["DepartmentType"]);
string codeDoc = properties.Web.Lists[new Guid(lookupDep.LookupList)].GetItemById(itemIDDoc)["CodeConfiguration"].ToString();

li[li.Fields.GetFieldByInternalName("CodeDoc").Id] = codeYear + "-" + codeLang + "-" + codeDep + "-" + codeDoc;
li[li.Fields.GetFieldByInternalName("PublishingDate").Id] = localDate;
li[SPBuiltInFieldId.FileLeafRef] = values.Replace('.', '-') + "_" + li.ID;
li.SystemUpdate(false);

```

Ilustración 4.15. Código lógica añadir código de documento

En esta primera imagen podemos ver la lógica para poder añadir el código al documento. Para ello primero de todo se obtienen todos los campos que conforman el código mediante su nombre interno. Una vez obtenidos los campos se genera el código y se añade como la propiedad, de nombre "Codedoc" del ítem que se ha creado al subir el documento.

```

SPFieldUserValueCollection usersAccept = GetUsersFromItemList("UsersAccept", properties);
if (li[li.Fields.GetFieldByInternalName("UsersRead").Id] != null)
{
    SPFieldUserValueCollection usersRead = GetUsersFromItemList("UsersRead", properties);
    foreach (SPFieldUserValue user in usersRead)
    {
        usersAccept.Add(user);
    }
}

RemovePermissionsWithElevatedPrivileges(properties);
AssignPermissionsWithElevatedPrivileges(properties, usersAccept, "Reader");
SPFieldUserValueCollection promoters = GetUsersFromItemList("Promotor", properties);
AssignPermissionsWithElevatedPrivileges(properties, promoters, "Editor");

```

Ilustración 4.16 Código asignación permisos documento

En esta segunda imagen se encuentra la lógica relacionada con la asignación de permisos del documento. En la primera parte del código se recuperan los usuarios a los que se les han de conceder permisos para poder ver el documento. Una vez recuperados, se rompen los permisos del ítem del documento (vienen heredados por los permisos que tiene el sitio). Una vez rotos los permisos se eliminan los permisos antiguos y se añaden los usuarios que se han recuperado al inicio.

```

int currentUserID = web.CurrentUser.ID;
Guid siteid = site.ID;
Guid webid = web.ID;
Guid listid = list.ID;
int itemid = newItem.ID;

string logItem = "New document";

var guid = site.AddNewItem(Guid.NewGuid(), DateTime.Now.ToUniversalTime(), Worker.WorkItemType, webid, siteid, itemid, false, listid, Guid.Empty, currentUserID, null, logItem, Guid.Empty);

```

Ilustración 4.17. Código poner en cola un ítem

Por último solo queda enviar los correos a los usuarios, dado que este proceso puede tardar debido a que se puede seleccionar un número muy elevado de usuarios a los que se les enviará el correo, se utilizará un timer job que será el que se encargue de enviar todos los correos. Para ello se utiliza la función de `AddWorkItem` que nos permite poner en una cola un ítem para que una vez se ejecute el timer job procese todos los usuarios y les envíe el correo a todos ellos.

#### 4.5.6 Receptor documentos internos

La lógica de este receptor es la misma que se encuentra para el receptor de documentos controlados, a excepción de que en este caso no tenemos usuarios que tengan que marcar el leído y comprendido, por lo que no será necesario pasar el ítem al Timer Job.

#### 4.5.7 Timer Jobs

Un Timer Job, es otra de las herramientas que nos da SharePoint y es prácticamente igual que un Receptor de Evento, con la única diferencia que el Timer Job no se ejecuta al desencadenarse un evento, sino se programa para que se ejecute cada cierto tiempo. El tiempo en el que se ejecuta depende de cómo haya sido configurado por el usuario, ejecutándose cada vez que pase el intervalo de tiempo que se le haya configurado.

Para el proyecto se han utilizado dos Timer Jobs diferentes. Uno de ellos se utiliza para comprobar la caducidad de los documentos y está configurado para que se ejecute una vez, cada día. El otro, es un Timer Job diferente, está configurado para que se ejecute cada hora, pero en este caso es un Timer Job de tipo Worker. Este Timer Job tiene la particularidad que, al ejecutarse, tiene una cola de elementos que procesar, al iniciarse, mira si en su cola tiene algún elemento pendiente de procesar, en caso positivo ejecuta todos los elementos que tenga pendientes. Este Timer Job es utilizado para poder enviar los correos a los usuarios que se añaden en el receptor que se ha mencionado antes en el punto anterior.

Para la creación de un Timer Job hay que añadir un nuevo elemento en la solución, de tipo clase. Una vez se haya creado la nueva clase, hay que añadir la estructura de una clase de un Timer Job. Para ello hay que añadir la definición *SPJobDefinition* para representar el tipo de clase que crearemos y añadir los tres constructores de la clase.

```
class TimerDemo : SPJobDefinition
{
    0 references | Sergi Galceran Serrano, 137 days ago | 1 author, 1 change
    public TimerDemo() : base() { }
    0 references | Sergi Galceran Serrano, 137 days ago | 1 author, 1 change
    public TimerDemo(string jobName, SPService service, SPServer server, SPJobLockType targetType) : base(jobName, service, server, targetType) { }
    1 reference | Sergi Galceran Serrano, 137 days ago | 1 author, 1 change
    public TimerDemo(string jobName, SPWebApplication webApplication) : base(jobName, webApplication, null, SPJobLockType.ContentDatabase)
    {
        this.Title = jobName;
    }
}
```

Ilustración 4.18. Estructura de un TimerJob

Una vez creada la clase con la lógica relativa al Timer Job, hay que crear una nueva Feature, que se asociará a esta nueva clase que se ha construido. En esta Feature será donde se configurará el tiempo de ejecución (cada cuanto se ejecuta). Para crear esta Feature se agregará otro ítem de tipo Feature. Una vez creada la nueva Feature, habrá que añadirle un evento receiver, para ello hay que hacer clic con el botón derecho y nos dará la opción. Una vez creada, tendremos una clase con métodos comentados, cada uno de estos métodos son receptores de eventos que corresponden con una acción asociada a la Feature, como cuando se activa la Feature o se desactiva. Los métodos que nos interesan son `FeatureActivated` que salta cuando la Feature es activada y `FeatureDeactivating` que se ejecuta al desactivar la Feature. Adicionalmente a estos métodos, crearemos dos funciones nuevas donde pondremos la lógica que se aplicara al activar la Feature y al desactivarla, estos métodos se encargaran de ejecutar la lógica asociada en la clase que se ha creado antes y, hacer que pare de ejecutarse nuestra lógica una vez se desactiva la Feature.

```
public class TimerJobDemoEventReceiver : SPFeatureReceiver
{
    const string JobName = "Demo Job";
    1 reference | Sergi Galceran Serrano, 138 days ago | 1 author, 1 change
    public override void FeatureActivated(SPFeatureReceiverProperties properties)
    {
        try
        {
            SPSecurity.RunWithElevatedPrivileges(delegate ()
            {
                SPWebApplication parentWebApp = (SPWebApplication)properties.Feature.Parent;
                DeleteExistingJob(JobName, parentWebApp);
                CreateJob(parentWebApp);
            });
        }
        catch (Exception ex)
        {
            throw ex;
        }
    }
}
```

**Ilustración 4.19 Código activación del TimerJob**

En esta primera parte al activar la Feature en SharePoint eliminamos otro Timer Job de esta clase que se estuviera ejecutando y creamos uno de nuevo. Se define un nombre de Job, para que posteriormente se pueda recuperar por ese nombre y borrarlo.



```

private void CreateJob(SPWebApplication site)
{
    try
    {
        TimerJobs.DocumentExpiration job = new TimerJobs.DocumentExpiration(JobName, site);
        //Execute in 3:00 OClock
        SPDailySchedule spDailySchedule = new SPDailySchedule();
        spDailySchedule.BeginHour = 04;
        spDailySchedule.BeginMinute = 00;
        spDailySchedule.BeginSecond = 0;
        spDailySchedule.EndHour = 05;
        spDailySchedule.EndMinute = 10;
        spDailySchedule.EndSecond = 0;
        job.Schedule = spDailySchedule;
        job.Update();
    }
    catch (Exception)
    {
        //return jobCreated;
    }
}

```

Ilustración 4.20 Código creación TimerJob

La lógica para crear un nuevo Timer Job, se define el tiempo cada cuanto se ejecutará y se crea un Timer Job, que se asocia con una clase en concreto como la que se ha creado en el anterior apartado. En el caso de los Timers Job de caducidad fueron programados para ejecutarse una vez al día, entre las 4 y las 5 de la mañana.

En esta parte se controla que al desactivar la feature asociada al timer, este deje de ejecutarse. Para ello se llama a la función de *DeleteExistingJob* que es donde se pondrá la lógica para ello.

```

public bool DeleteExistingJob(string jobName, SPWebApplication site)
{
    bool jobDeleted = false;
    try
    {
        foreach (SPJobDefinition job in site.JobDefinitions)
        {
            if (job.Name == jobName)
            {
                job.Delete();
                jobDeleted = true;
            }
        }
    }
    catch (Exception)
    {
        return jobDeleted;
    }
    return jobDeleted;
}

```

Ilustración 4.21 Código eliminación TimerJob existente

En esta parte se elimina el timer que se ha creado, para ello se borra timer pasandole como parametro el nombre de este.

#### 4.5.8 Timer Job de Caducidad de Documentos

En el proyecto hay dos TimerJob de este tipo, uno para documentos internos y otro para controlados. El funcionamiento de estos los mismos es de obtener los días de aviso antes que un documento caduque en la lista de configuración, y después obtener todos los documentos que pasados los días que se han obtenido de la configuración, contando desde ese día, coincida con la fecha de caducidad de ese documento. La lógica utilizada en el proyecto es la misma para las dos bibliotecas, con la única diferencia que cada uno de los TimerJob busca en una de las bibliotecas.

```
//string siteUrl = this.WebApplication.Sites[0].Url;
using (SPSite site = new SPSite("https://intranet.lacer.es/sites/gestordocumental"))
{
    using (SPWeb web = site.OpenWeb())
    {
        base.Execute(targetInstanceId);
        SPList docControllist = web.Lists.TryGetList("Documentos Internos");
        string reminderDay = GetConfigurationValue(web, "RDE");
        int reminderDays = Convert.ToInt32(reminderDay);
        SPListItemCollection items = GetNotificationDocumentsDueDate(docControllist, reminderDays);
        CaducityDocumentReminder(items, web);
    }
}
```

Ilustración 4.22 Código obtención días de la lista de configuración

Primero se obtienen los dias de la lista de configuración.

```
public string GetConfigurationValue(SPWeb web, string code)
{
    string configValue = "";
    SPList list = web.Lists.TryGetList("Configuracion");

    SPListItemCollection listcollection = list.GetItems(new SPQuery()
    {
        Query = string.Format("<Where><Eq><FieldRef Name='CodeConfiguration'/><Value Type='Text'>{0}</Value></Eq></Where>", code)
    });

    if (listcollection.Count > 0)
        configValue = listcollection[0][listcollection[0].Fields.GetFieldByInternalName("ValueConfiguration").Id].ToString();

    return configValue;
}
```

Ilustración 4.23. Código obtener documentos caducados

Una vez se obtienen los dias obtenemos los documentos que cumplan con la condición de caducidad. Para ello se obtienen a través de una caml query [2] que nos permite obtener los items de una lista que cumplen las características que le indicamos en la query.

```

public SPListItemCollection GetNotificationDocumentsDueDate(SPList list, int offsetDays)
{
    SPQuery caml = new SPQuery
    {
        Query =
        @"<Where>
            <And>
                <Eq>
                    <FieldRef Name='Validity' />
                    <Value Type='Boolean'>1</Value>
                </Eq>
                <Eq>
                    <FieldRef Name='ValidityDate' />
                    <Value Type='DateTime' IncludeTimeValue='False'><Today OffsetDays=''' + offsetDays + ''' /></Value>
                </Eq>
            </And>
        </Where>"
    };
    SPListItemCollection item = list.GetItems(caml);
    return item;
}

```

**Ilustración 4.24. Caml Query obtención de documentos activos y caducados**

Por último se obtienen todos los promotores asociados de los documentos que ha devuelto la query anterior, y se envía un correo a todos ellos.

```

public void CaducityDocumentReminders(SPListItemCollection remindersDocs, SPweb web)
{
    foreach (SPListItem document in remindersDocs)
    {
        string itemFilter = document["DocumentName"].ToString();
        string smtpAddress = getSMTPAddress(web.Url);
        SPFieldUserValueCollection users = GetUsersFromItemList("Promotor", document, web);

        var listUsers = ResolveUsersFromItemList(web, users);
        foreach (SPPrincipalInfo promotor in listUsers)
        {
            if (!string.IsNullOrEmpty(promotor.Email))
            {
                bool docControl = document.Url.Contains("Documentos Internos");
                string[] mailInfo = GetMailConfigurationValue(web, "CMC");
                string subject = mailInfo[0].Replace("@@NameDoc@@", document["DocumentName"].ToString());
                string contentLink = mailInfo[1].Replace("@@link@@", "<a href='\" + web.Url + \"/lists/Documentos%20Controlados/forms/DispForm.aspx?ID=" + document.ID + "'> link</a>");
                string content = contentLink.Replace("@@NameDoc@@", document["DocumentName"].ToString());
                SendEmail(smtpAddress, "intranet@lacer.es", promotor.Email, subject, content, true);
                document.Update();
            }
        }
    }
}

```

**Ilustración 4.25. Código obtención configuración del correo**

## 4.5.9 Timer Job de Envío de Correos

Este es otro tipo de Timer Job que hay en el proyecto. Como se ha comentado antes, este Timer Job es el encargado de enviar los correos de los documentos pendientes a firmar a los usuarios.

La lógica para crear este tipo de Timer Job es muy igual que la utilizada para crear un Timer Job, pero al crear la clase en este Timer Job en concreto heredaremos de la clase *SPWorkItemJobDefinition*.

Para poder acceder a todos los items que se encuentran en la cola hay que usar el método *ProcessWorkItem*, que permite ir obteniendo los ítems que se han ido guardando en él.

```

public class Worker : SPWorkItemJobDefinition
{
    public static readonly string WorkItemJobDisplayName = "Workitem Job";
    public static readonly Guid WorkItemTypeId = new Guid("{CEAAFFA4-4391-40D6-868E-19EDED87DD4}");
}

```

Ilustración 4.26 Definición del TimerJob Worker

```

using (SPSite site = new SPSite(workItem.SiteId))
{
    using (SPWeb web = site.OpenWeb(workItem.WebId))
    {
        try
        {
            string smtpAddress = getSMTPAddress(web.Url);
            SPList list = web.Lists.TryGetList("Estado Documentos");
            SPListItem listItem = list.GetItemById(workItem.ItemId);
            UpdateListDoc(web, listItem);
            SPFieldUserValueCollection userSign = GetUsersFromItemList("UserSign", listItem);
            var listUsers = ResolveUsersFromItemList(web, userSign);
            string mail = listUsers[0].Email;
            string displayFormUrl = site.MakeFullUrl(listItem.ParentList.DefaultDisplayFormUrl);
            string[] mailInfo = GetMailConfigurationValue(web, "CID");
            //string documentUrl = string.Format("{0}?ID={1}", displayFormUrl, listItem["DocumentId"].ToString());
            string docName = listItem["DocumentName"].ToString();
            string docId = listItem["DocumentId"].ToString();
            string subject = mailInfo[0].Replace("@@nameDoc@", docName);
            string bodyOld = mailInfo[1].Replace("@@link@", "<a href=\"" + web.Url + "/Lists/Documentos%20Controlados/Forms/DispForm.aspx?ID=" + docId + "\"> link</a>");
            string body = bodyOld.Replace("@@nameDoc@", listItem["DocumentName"].ToString());
            bool mailSent = SendEmail(smtpAddress, "intranet@lacer.es", mail, subject, body, true);
            listItem.SystemUpdate(false);
        }
        catch (Exception ex)
        {
            // exception handling
            SPListItem newItem = web.Lists["Log"].AddItem();
            if (!string.IsNullOrEmpty(ex.Message))
            {
                newItem[SPBuiltinFieldId.Title] = ex.Message.Substring(0, 200);
            }
            else
            {
                newItem[SPBuiltinFieldId.Title] = "ERROR";
            }
            newItem.Update();
        }
        finally
        {
            // SubCollection(): required to set the ParentWeb property on the WorkItemsCollection
            // Delete(): required to remove the item from the queue
            workItems.SubCollection(site, web, 0, (uint)workItems.Count).DeleteWorkItem(workItem.Id);
        }
    }
}

```

Ilustración 4.27 Código WorkItem de envío de correos

Una vez se obtiene el ítem, se accede al campo del ítem con los usuarios a los que se les ha de enviar el correo. Obtenemos los valores de configuración del correo de la lista de configuración y se envía el correo.

#### 4.5.10 Timer Job de Recordatorios

Este es el último Timer Job que se utiliza en el proyecto, siendo el encargado de enviar correos de recordatorio según los parámetros que se le indiquen en la lista de configuración. Para ello comprueba todos los documentos que están pendientes de firma y comprueba cuanto tiempo ha pasado en días, después comprueba si el número de días que ha pasado coincide con el parámetro de la lista de configuración, que determina cada cuanto se envían los recordatorios, en los documentos que coincida se procede al envío del correo. Se define un máximo de recordatorios también configurable desde la lista de configuración.

```
string smtpAddress = getSMTPAddress(web.Url);
SPList docControllist = web.Lists.TryGetList("Documentos Controlados");
SPList stateDocuments = web.Lists.TryGetList("Estado Documentos");
SPListItemCollection vigentDocs = GetVigentDocuments(docControllist);
string numberReminders = GetConfigurationValue(web, "RE");
string reminderDays = GetConfigurationValue(web, "RD");
```

**Ilustración 4.28 Obtención valores de la lista de configuración**

En la primera parte de este timer se obtienen los valores de la lista de configuración RE (número de recordatorios) y RD (días que han de transcurrir para enviar un recordatorio).

```
SPListItemCollection docsRemind = GetNotificationReminders(stateDocuments, newName, numberReminders, reminderDays);
foreach (SPListItem docRemind in docsRemind)
{
    try
    {
        int numReminderDoc = Convert.ToInt32(docRemind["NumberReminderSign"].ToString());
        int numReminder = Convert.ToInt32(numberReminders);
        docRemind["NumberReminderSign"] = (numReminderDoc + 1).ToString();
        docRemind["LastReminderSignDate"] = DateTime.Now;
        if (docRemind["UserSign"].ToString() != null)
        {
            SPFieldUserValueCollection usersSign = GetUsersFromItemList("UserSign", docRemind);
            var listUsers = ResolveUsersFromItemList(web, userSign);
            string mailReminder = listUsers[0].Email;
            //string nombreDoc = docRemind["DocumentName"].ToString();
            if (numReminder == numReminderDoc + 1)
            {
                SPFieldUserValueCollection promoters = GetUsersFromItemList("Promotor", doc);
                //docRemind["DocumentName"] = "Último aviso ";
                string[] mailInfo = GetMailConfigurationValue(web, "CMUR");
                string subject = mailInfo[0].Replace("@@nameDoc@", docRemind["DocumentName"].ToString());
                string contentLink = mailInfo[1].Replace("@@link@", "<a href='" + web.Url + "/Lists/Documentos%20Controlados/Forms/DispForm.aspx?ID=" + docRemind.ID + "'> link</a>");
                string content = contentLink.Replace("@@nameDoc@", docRemind["DocumentName"].ToString());
                bool mailSent = SendEmail(smtpAddress, "intranet@lacer.es", mailReminder, subject, content, true);
                string[] mailInfoP = GetMailConfigurationValue(web, "CMFP");
                string subjectP = mailInfoP[0].Replace("@@nameDoc@", docRemind["DocumentName"].ToString());
                string contentPLink = mailInfoP[1].Replace("@@link@", "<a href='" + web.Url + "/Lists/Documentos%20Controlados/Forms/DispForm.aspx?ID=" + docRemind.ID + "'> link</a>");
                string contentP = contentPLink.Replace("@@nameDoc@", docRemind["DocumentName"].ToString());
                foreach (SPFieldUserValue promoter in promoters)
                {
                    if (promoter.User.Email != null)
                    {
                        bool mailSentPro = SendEmail(smtpAddress, "intranet@lacer.es", promoter.User.Email, subjectP, contentP, true);
                    }
                }
            }
            else
            {
                string[] mailInfo = GetMailConfigurationValue(web, "CMR");
                string subject = mailInfo[0].Replace("@@nameDoc@", docRemind["DocumentName"].ToString());
                string contentLink = mailInfo[1].Replace("@@link@", "<a href='" + web.Url + "/Lists/Documentos%20Controlados/Forms/DispForm.aspx?ID=" + docRemind.ID + "'> link</a>");
                string content = contentLink.Replace("@@nameDoc@", docRemind["DocumentName"].ToString());
                bool mailSent = SendEmail(smtpAddress, "intranet@lacer.es", mailReminder, subject, content, true);
            }
            docRemind.Update();
        }
    }
    catch (Exception e)
    {
        //
    }
}
```

**Ilustración 4.29 Código envío correos del Worker**

Una vez se obtienen los valores anteriores, se buscan todos los documentos pendientes de firmar. Estos documentos se procesan y se mira los recordatorios que se han enviado, si no ha llegado al máximo se envía otro correo al usuario y se incrementa en uno el número de recordatorios de ese documento. El texto que se envía al usuario se obtiene de una lista de configuración. En caso de que sea el último recordatorio que se va a enviar (ha llegado al número máximo de recordatorios) se envía un correo igual al usuario informándole que este es último recordatorio que recibirá y adicionalmente se enviará un correo al promotor del documento informándole que usuario tiene pendiente de firmar el documento.

### 4.5.11 Estructura final del proyecto e instalación

En este punto del proyecto, se ha creado toda la estructura del proyecto, solo queda trasladarlo a SharePoint. Para ello se debe generar un paquete, que posteriormente, mediante unos comandos se instalará en SharePoint.

La solución se crea a partir de las features que tenemos en una solución. Para este proyecto se han creado dos soluciones diferentes, una contiene todos los Timer Jobs, y el otro contiene la estructura del proyecto con todos los elementos (listas, bibliotecas, columnas, etc ...). Esta separación es debido a que en un futuro, en caso de querer volver a instalar este proyecto, haya una separación entre la estructura y los Timer Jobs, de tal manera que se puede instalar uno sin dependencia del otro.

Dentro de la solución se encuentran diferentes features, que al instalar la solución en SharePoint, se pueden desactivar de manera individual, es decir, en una solución se pueden activar o desactivar features según se requiera. En este caso en concreto para la solución de que contiene la estructura, los elementos están agrupados por feature, por lo que hay cinco features. En el caso de los timer jobs sucede el mismo caso.

La feature se genera automáticamente al añadir elementos a la solución, o se puede crear haciendo clic derecho con el ratón en la carpeta de features (se genera al crear la solución de SharePoint) y si hacemos clic en ella se puede seleccionar que ítems de los que hay creados en la solución se asociaran.

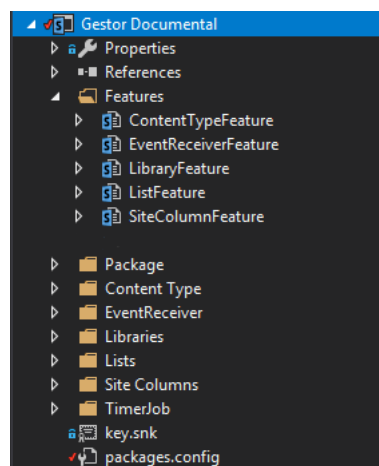
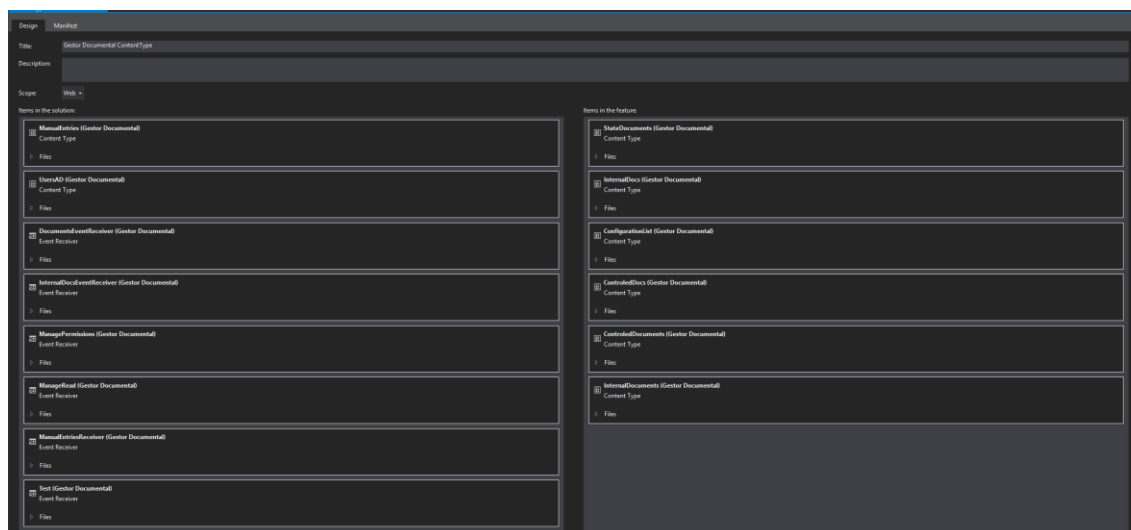


Ilustración 4.30 Estructura feature

En la imagen podemos ver un ejemplo de la feature de tipos de contenido. A la derecha se encuentran los ítems que se pueden añadir a la feature, a la derecha aquellos que están añadidos.



**Ilustración 4.31. Pantalla de la feature**

Una vez creadas las features queda generar el paquete que después puede ser instalado en un entorno de SharePoint. Para generar el paquete hay que hacer clic en la solución y marcar la opción de publish, esto hará que se abra una ventana donde hay que seleccionar donde se guarda el paquete que se genere.

El último paso corresponde al despliegue del paquete en SharePoint, para ello hay que ejecutar los comandos en un ordenador con acceso a la configuración del SharePoint donde se quiera desplegar. Una vez se acceda al ordenador se han de ejecutar dos comandos

```
Add-SPSolution -LiteralPath c:\solucion.wsp
```

En este primer comando se añade el paquete que, posteriormente se instala. Simplemente hay que pasarle la ruta del paquete que se ha generado con Visual Studio. Una vez ejecutado, ya está listo para ser instalado, por lo que hay que ejecutar el segundo comando que instalará el paquete que se acaba de añadir.

```
Install-SPSolution -Identity contoso_solution.wsp -GACDeployment
```

En este caso se pasa como parámetro el nombre del paquete y hay que añadirle la opción de -GACDeployment que indica que el paquete que se instala es de confianza. Una vez se ha ejecutado el comando, el paquete está instalado en SharePoint.

En esta última parte del proyecto se modificó la estructura que se ha desplegado en el apartado anterior. En esta primera parte se ha añadido un TypeScript en la biblioteca de controlados, que cuando un usuario tiene



pendiente de leer y comprender, cuando hace clic al documento lo redirige a una página diferente donde le da la opción de darle al leído y comprendido.

```
(function () {
    var overrideNameField = {};
    overrideNameField.Templates = {};
    overrideNameField.Templates.OnPostRender = hiddenFieldOnPreRender;
    overrideNameField.Templates.Fields = {
        "DetailDoc": { "View": overrideDetailDocFieldTemplate, "DisplayForm": hiddenFieldTemplate, "EditForm": hiddenFieldTemplate },
        "DepartmentType": { "View": overrideLookupFieldTemplate, "DisplayForm": overrideDispLookupFieldTemplate },
        "DocType": { "View": overrideLookupFieldTemplate, "DisplayForm": overrideDispLookupFieldTemplate },
        "LanguageDoc": { "View": overrideLookupFieldTemplate, "DisplayForm": overrideDispLookupFieldTemplate },
        "Companies": { "View": overrideLookupFieldTemplate, "DisplayForm": overrideDispLookupFieldTemplate },
        "DocumentName": { "View": overrideNameFieldTemplate, "EditForm": hiddenFieldTemplate },
        "UsersAccept": { "DisplayForm": hiddenFieldTemplate },
        "UsersRead": { "DisplayForm": hiddenFieldTemplate },
        "FileLeafRef": { "DisplayForm": hiddenFieldTemplate, "EditForm": hiddenFieldTemplate },
        "StateSign": { "DisplayForm": hiddenFieldTemplate, "View": overrideStateSignFieldTemplate, "EditForm": hiddenFieldTemplate },
        "Estado": { "View": overrideStateSignFieldTemplate, "DisplayForm": hiddenFieldTemplate, "EditForm": hiddenFieldTemplate },
        "Tags_x0020_Documentos_x0020_Corporativos": { "DisplayForm": hiddenFieldTemplate }, //, "View": overrideDispLookupFieldTemplate }
        "ReadOnlyLink": { "DisplayForm": hiddenFieldTemplate },
        "Documento_x0020_Externos": { "DisplayForm": hiddenFieldTemplate },
    };
});
```

### Ilustración 4.32 Javascript desactivación link ítems

```
function overrideDetailDocFieldTemplate(ctx) {
    var tmplText = "";
    //var url = _spPageContextInfo.webServerRelativeUrl + '_api/web/lists(' + _spPageContextInfo.pageListId + ')/items(' + ctx.CurrentItem['ID'] + ')/hasuniqueoleassignments';
    //var url = _spPageContextInfo.webServerRelativeUrl + '_api/web/lists(' + _spPageContextInfo.pageListId + ')/items(' + ctx.CurrentItem['ID'] + ')/effectiveBasePermissions';
    var url = _spPageContextInfo.webServerRelativeUrl + '_api/web/lists(' + ctx.ListName + ')/items(' + ctx.CurrentItem['ID'] + ')/effectiveBasePermissions';

    var placeholderId = ctx.CurrentItem.ID;

    if (ctx.ListTitle.toLowerCase() == "documentos controlados") {
        if (ctx.CurrentItem["CodeDoc"] != null && ctx.CurrentItem["CodeDoc"] != "") {
            var title = ctx.CurrentItem.Title;
            var fileRef = ctx.CurrentItem["FileRef"];
            var fileLeafRef = ctx.CurrentItem["FileLeafRef"];
            var fileDocumentId = ctx.CurrentItem["ID"];

            var fileName = fileLeafRef.substring(0, fileLeafRef.lastIndexOf("."));
            var detailUrl = _spPageContextInfo.webServerRelativeUrl + "/Lists/Estado Documentos/Peticiones/200Documento.aspx?FilterField1=DocumentId&FilterValue1=" + fileDocumentId;

            tmplText = "<a href='" + detailUrl + "' target='_blank'>Ver aceptaciones</a>";
        }
    }

    // Make the AJAX call.
    $.ajax({
        url: url,
        type: "GET",
        headers: { 'Accept': 'application/json;odata=verbose' },
        success: function (data) {
            SP.SOD.executeFunc('SP.js', 'SP.ClientContext', function () {
                var manageListsPerms = new SP.BasePermissions();
                manageListsPerms.fromJson(data.d.EffectiveBasePermissions);
                var manageLists = manageListsPerms.has(SP.PermissionKind.editListItems);

                if (manageLists) {
                    $('#DetailDoc' + placeholderId).html(tmplText);
                }
            });
        }
    });

    // Return the placeholder for the async AJAX call.
    return "<div id='DetailDoc' + placeholderId + '>' />";
}
```

### Ilustración 4.33 Javascript redirección al formulario custom

Para agregar este TypeScript hay que entrar en la biblioteca donde se quiera añadir y se edita la página. Para poder editar la página hay que hacer clic en la rueda de configuración que se encuentra en la parte superior derecha de la página y en el menú seleccionar editar página. Aparece la misma pantalla que había antes, pero se ha dividido en secciones, cada una de estas secciones es un elemento web que se puede editar, en este caso queremos que cuando un usuario haga clic en el nombre del documento y lo tiene pendiente de aceptar el leído y comprendido lo redirija. Por ello hay que editar el segundo elemento web, haciendo clic en el punto que aparece al lado del cuadrado y haciendo clic en editar elemento web del menú que aparecerá. Ahora aparece otro recuadro con las propiedades del elemento web, hay que ir al apartado de varios y donde aparece vínculo JavaScript se puede añadir una ruta de dentro del SharePoint donde subiremos nuestro fichero que contiene el JavaScript, en el proyecto este archivo se guardó en la carpeta de Style Library, dentro de una carpeta que pone la compañía.



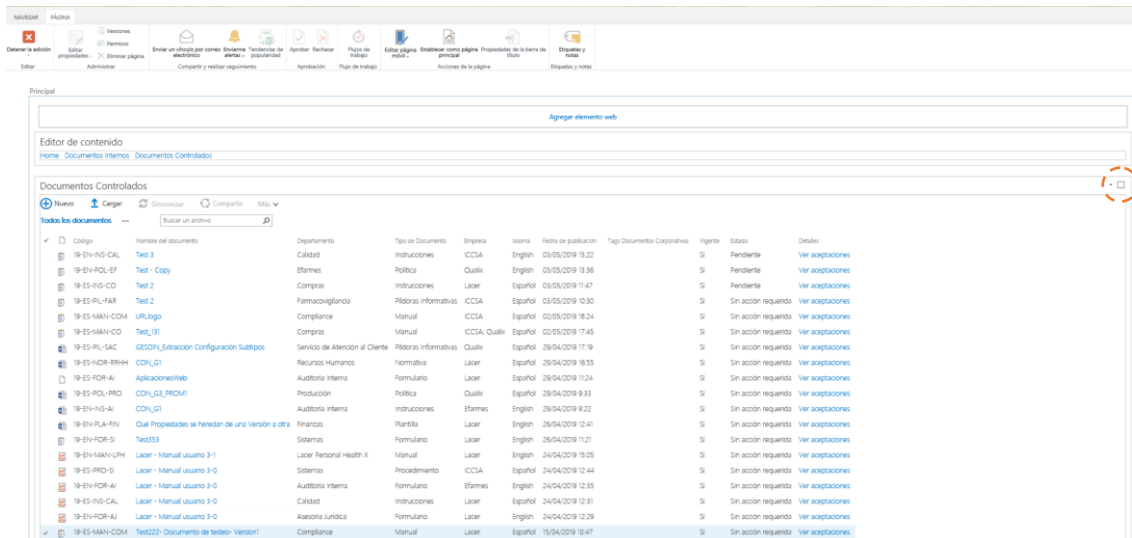


Ilustración 4.34 Página en modo edición

Documentos Controlados

Mostrar vistas

Puede editar la vista actual o seleccionar otra vista.

Vista seleccionada

<Vista actual>

Editar la vista actual

Tipo de barra de herramientas

Barra de herramientas completa

Apariencia

Distribución

Avanzado

Opciones de AJAX

Varios

Datos de ejemplo

Vínculo XSL

☒ Habilitar almacenamiento en caché de vista de datos

Tiempo de espera de de almacenamiento en caché de la vista de datos (segundos)

86400

☒ Enviar la primera fila a los elementos web conectados cuando se cargue la página

☐ Representación del servidor

☐ Deshabilitar el menú de selector de vistas

☒ Deshabilitar el botón de guardar esta vista

☒ Mostrar cuadro de búsqueda

Vínculo JS

~sitecollection/Style Library/Lt

Aceptar Cancelar Aplicar

Ilustración 4.35 Formulario edición de la webpart

## CAPÍTULO 5. Solución final

En este apartado se muestra el resultado final de la implementación de la solución en el SharePoint de la compañía, así como su funcionamiento.

### 5.1 Documentos internos

Esta biblioteca es la de documentos internos, para poder agregar un nuevo documento hay que hacer clic en el botón de nuevo y esto abre dos formularios, primero nos aparece uno para subir el documento (Ilustración 5.2) y el después otro para rellenar sus propiedades (Ilustración 5.3).

#### Documentación Oficial Compañía - Documentos Internos ⓘ

[Home](#) [Documentos Internos](#) [Documentos Controlados](#)

[Nuevo](#) [Cargar](#) [Sincronizar](#) [Compartir](#) [Más](#) ▾

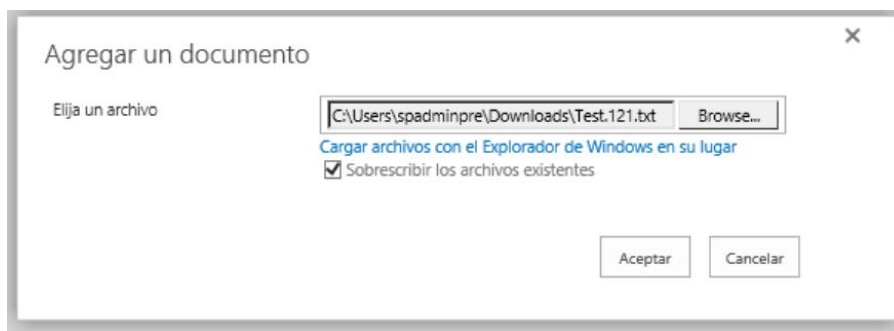
Todos los documentos ...

Buscar un archivo

✓	Código	Nombre del documento	Tipo de Documento	Departamento	Empresa	Idioma	Fecha de publicación	Tags Documentos Corporativos	Vigente
	19-ES-MAN-CO	Test-121	Manual	Compras	ICCISA; Efarmes	Español	15/04/2019 10:27		Sí
	19-ES-NOR-AI	tst	Normativa	Auditoría Interna	Efarmes	Español	09/04/2019 17:36	Acoso; Bullying;	Sí

Arrastre archivos aquí para cargar

Ilustración 5.1 Biblioteca Documentos Internos



Agregar un documento

Elija un archivo

C:\Users\spadminpre\Downloads\Test.121.txt [Browse...](#)

[Cargar archivos con el Explorador de Windows en su lugar](#)

☒ Sobrescribir los archivos existentes

[Aceptar](#) [Cancelar](#)

Ilustración 5.2 Formulario carga documento

Agregar un documento

EDITAR

Guardar Cancelar Pegar Copiar Eliminar elemento Revisión ortográfica

Ejecutar Portapapeles Acciones Revisión ortográfica

Proteger Cancelar

Código

Departamento \* (Ninguno)

Tipo de Documento \* (Ninguno)

Idioma \* (Ninguno)

Empresa \*

Efarmes  
ICCSA  
Lacer  
Qualix

Agregar >  
< Quitar

Personas con acceso lectura \*

Especifique los usuarios separados por signos de punto y coma.

Promotor \*

Especifique los usuarios separados por signos de punto y coma.

Tags Documentos Corporativos

Fecha de caducidad

Vigente ☒

Creado el 10/06/2019 22:24 por Oscar Salobreña  
Última modificación realizada el 10/06/2019 22:24 por Oscar Salobreña

Proteger Cancelar

**Ilustración 5.3 Formulario propiedades del documento**

Una vez rellenos los campos se sube el nuevo documento con las propiedades rellenas. En caso de no rellenar los campos obligatorios (marcados con asterisco) no deja guardar el documento aparecerán marcados en el formulario los campos que faltan por rellenar (Ilustración 5.4).

Agregar un documento

EDITAR

Guardar

Cancelar

Pegar

Copiar

Eliminar elemento

Revisión ortográfica

Código

Departamento \*

Tipo de Documento \*

Idioma \*

Empresa \*

Personas con acceso lectura \*

Promotor \*

Tags Documentos Corporativos

Fecha de caducidad

Vigente

(Ninguno)

(Ninguno)

(Ninguno)

Efarmes  
ICCSA  
Lacer  
Qualix

Debe especificar un valor para este campo obligatorio.

Debe especificar un valor para este campo obligatorio.

Debe especificar un valor para este campo obligatorio.

Debe especificar un valor para este campo obligatorio.

Especifique los usuarios separados por signos de punto y coma.

Especifique los usuarios separados por signos de punto y coma.

Agregar >

< Quitar

Proteger

Cancelar

Creado el 10/06/2019 22:30 por Oscar Salobreña

Última modificación realizada el 10/06/2019 22:30 por Oscar Salobreña

Proteger

Cancelar

Ilustración 5.4 Error al no rellenar los campos

5.2 Documentos controlados

El funcionamiento de esta biblioteca es idéntico al de documentos internos, hay que hacer clic en el botón de nuevo y nos aparecen dos formularios, al igual que en el caso anterior. En este caso en el segundo formulario (Ilustración 5.6) cambian algunas propiedades.

Documentación Oficial Compañía - Documentos Controlados

Home Documentos Internos Documentos Controlados

Nuevo

Cargar

Sincronizar

Compartir

Más

Todos los documentos

Buscar un archivo

	Código	Nombre del documento	Departamento	Tipo de Documento	Empresa	Idioma	Fecha de publicación	Tags Documentos Corporativos	Vigente
	19-EN-INS-CAL	Test 3	Calidad	Instrucciones	ICCSA	English	03/05/2019 15:22		Si
	19-EN-POL-EF	Test - Copy	Efarmes	Politica	Qualix	English	03/05/2019 13:36		Si
	19-ES-INS-CO	Test 2	Compras	Instrucciones	Lacer	Español	03/05/2019 11:47		Si
	19-ES-PIL-FAR	Test 2	Farmacovigilancia	Pildoras Informativas	ICCSA	Español	03/05/2019 10:30		Si
	19-ES-MAN-COM	URLlogo	Compliance	Manual	ICCSA	Español	02/05/2019 18:24		Si
	19-ES-MAN-CO	Test_131	Compras	Manual	ICCSA, Qualix	Español	02/05/2019 17:45		Si
	19-ES-PIL-SAC	GESDIN_Extracción Configuración Subtipos	Servicio de Atención al Cliente	Pildoras Informativas	Qualix	Español	29/04/2019 17:19		Si
	19-ES-NOR-RRHH	CON_G1	Recursos Humanos	Normativa	Lacer	Español	29/04/2019 16:55		Si
	19-ES-FOR-AI	AplicacionesWeb	Auditoria Interna	Formulario	Lacer	Español	29/04/2019 11:24		Si
	19-ES-POL-PRO	CON_G3_PROMI	Producción	Politica	Qualix	Español	29/04/2019 9:33		Si
	19-EN-INS-AI	CON_G1	Auditoria Interna	Instrucciones	Efarmes	English	29/04/2019 9:22		Si
	19-EN-PLA-FIN	Qué Propiedades se heredan de una Versión a otra	Finanzas	Plantilla	Lacer	English	26/04/2019 12:41		Si
	19-EN-FOR-SI	Test1353	Sistemas	Formulario	Lacer	English	26/04/2019 11:21		Si
	19-EN-MAN-LPH	Lacer - Manual usuario 3-1	Lacer Personal Health X	Manual	Lacer	English	24/04/2019 15:05		Si
	19-ES-PRO-SI	Lacer - Manual usuario 3-0	Sistemas	Procedimiento	ICCSA	Español	24/04/2019 12:44		Si

Ilustración 5.5 Biblioteca documentos controlados

The screenshot shows the 'Agregar un documento' (Add a document) form. At the top, there is a toolbar with icons for 'Guardar' (Save), 'Cancelar' (Cancel), 'Pegar' (Paste), 'Copiar' (Copy), 'Eliminar elemento' (Remove element), and 'Revisión ortográfica' (Spelling check). Below the toolbar, the form contains several fields: 'Código' (empty), 'Departamento' (dropdown menu showing '(Ninguno)'), 'Tipo de Documento' (dropdown menu showing '(Ninguno)'), 'Idioma' (dropdown menu showing '(Ninguno)'), 'Empresa' (list box showing 'Efarmes', 'ICCSA', 'Lacer', and 'Qualix'), 'Personas de leído y comprendido' (text input), 'Personas con acceso lectura' (text input), 'Promotor' (text input), 'Tags Documentos Corporativos' (text input), and 'Fecha de caducidad' (calendar icon). At the bottom, there is a 'Vigente' checkbox which is checked, and a status bar showing 'Creado el 10/06/2019 22:34 por Oscar Salobreña' and 'Última modificación realizada el 10/06/2019 22:34 por Oscar Salobreña'. There are 'Proteger' and 'Cancelar' buttons at the top right and bottom right.

**Ilustración 5.6 Formulario propiedades documentos controlados**

The screenshot shows the 'Agregar un documento' form with validation errors. The 'Departamento' dropdown is set to 'Auditoría Interna', 'Tipo de Documento' is 'Instrucciones', and 'Idioma' is 'Español'. The 'Empresa' list box shows 'Efarmes', 'ICCSA', 'Lacer', and 'Qualix'. The 'Personas de leído y comprendido', 'Personas con acceso lectura', 'Promotor', and 'Tags Documentos Corporativos' fields are empty. The 'Fecha de caducidad' field has a calendar icon. The 'Vigente' checkbox is checked. The status bar shows 'Creado el 10/06/2019 22:34 por Oscar Salobreña' and 'Última modificación realizada el 10/06/2019 22:34 por Oscar Salobreña'. There are 'Proteger' and 'Cancelar' buttons at the top right and bottom right. Red error messages are displayed below the 'Empresa', 'Personas de leído y comprendido', 'Personas con acceso lectura', and 'Promotor' fields, stating 'Debe especificar un valor para este campo obligatorio.' (A value must be specified for this mandatory field).

**Ilustración 5.7 Error al no rellenar los campos**

También al igual que pasa con el otro formulario, en caso de no rellenar todos los campos obligatorios no nos deja guardar (Ilustración 5.7).

### 5.3 Listas de configuración

A continuación, se muestran las listas de configuración que hay en el proyecto con los diferentes parámetros de configuración de cada una de ellas.

En esta primera lista (Ilustración 5.8) se encuentran parámetros relacionados con las configuraciones de días y número de recordatorios.

#### Documentación Oficial Compañía - Configuración ⓘ

[+ nuevo elemento](#) o [modificar](#) esta lista

Todos los elementos

...

Buscar un elemento



✓	Abreviatura elemento	Valor del elemento	Descripción
	RE	3	Número de recordatorios
	RD	3	Días que han de transcurrir entre un recordatorio y el siguiente
	RDE	2	Días que han de transcurrir para enviar el aviso conforme el documento ha caducado

**Ilustración 5.8 Lista configuración recordatorios**

En esta segunda lista (Ilustración 5.9) se encuentra la información referente al envío de correos que se enviarán cuando haya un recordatorio.

#### Documentación Oficial Compañía - Configuración correos

[+ nuevo elemento](#) o [modificar](#) esta lista

Todos los elementos

...

Buscar un elemento



✓	Código Correo	Subject	Body
	CMD	[Aviso Gestor Documental] Nuevo documento de lectura obligatoria @nameDoc@	Por favor, lea el documento @nameDoc@ que se encuentra en el @link@ y confirme su lectura y comprensión.
	CMR	[Aviso Gestor Documental] Recordatorio de documento de lectura obligatoria @nameDoc@	Recuerde que tiene pendiente leer el documento @nameDoc@. Le agradeceríamos que proceda a su lectura en el @link@ con la mayor brevedad y que confirme su correspondiente lectura y comprensión.
	CMUR	[Aviso Gestor Documental] Último recordatorio de documento de lectura obligatoria @nameDoc@	Recuerde que tiene pendiente leer el documento @nameDoc@. Le informamos que es el último recordatorio que recibirá para este documento.
	CMFP	[Aviso Gestor Documental] Documentos no validados @nameDoc@	Le informamos que el documento @nameDoc@ no han sido validado por todos los usuarios requeridos para ello. Para acceder al aviso, pulse el @link@.
	CMC	[Aviso Gestor Documental] Documento a punto de caducar @nameDoc@	Le informamos que el documento @nameDoc@ está a punto de caducar. Para acceder al documento, pulse el @link@.

**Ilustración 5.9 Lista configuración correos**

En estas listas los usuarios administradores del espacio documental, pueden modificar todos los valores de la lista exceptuando el código o abreviatura.

En el segundo tipo de listas se encuentran listas que contienen la información de los campos de selección múltiple que estan al crear un nuevo documento. Estos campos son la empresa, el idioma y el departamento. En estas listas se pueden añadir o eliminar los elementos que hay en ellas.

## Documentación Oficial Compañía - Departamento ⓘ

[+ nuevo elemento](#) o [modificar](#) esta lista

[Todos los elementos](#) ...

Buscar un elemento 🔍

✓	Abreviatura	Nombre valor
	COM	Compliance
	AJ	Asesoría Jurídica
	AI	Auditoría Interna
	RRHH	Recursos Humanos
	FIN	Finanzas
	SI	Sistemas
	CON	Controlling
	FAR	Farmacovigilancia
	NF	Negocio Farma
	SG	Servicios Generales
	MK	Marketing
	REG	Registros
	LPH	Lacer Personal Health X
	CO	Compras
	LO	Logística

**Ilustración 5.10** Lista configuración departamentos

## Documentación Oficial Compañía - Empresa

[+ nuevo elemento](#) o [modificar](#) esta lista

[Todos los elementos](#) ...

Buscar un elemento 🔍

✓	Abreviatura	Nombre valor
	LA	Lacer
	EF	Efarnes
	IC	ICCSA
	QU	Qualix

**Ilustración 5.11** Lista configuración de empresas

## Documentación Oficial Compañía - Idioma

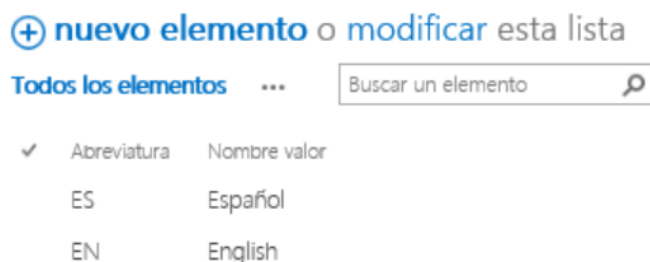


Ilustración 5.12 Lista configuración idioma

### 5.4 Caducidad de documentos

Todos los documentos, que al ser creados se les añade una fecha de caducidad en el formulario, esto hace que, según los días que haya configurados en la lista de configuración, se envíe un correo al promotor o promotores del documento. El texto que se envía también se obtiene de la lista de configuración de correos.

A continuación, se muestra un ejemplo del correo que recibiría el promotor (Ilustración 5.13).

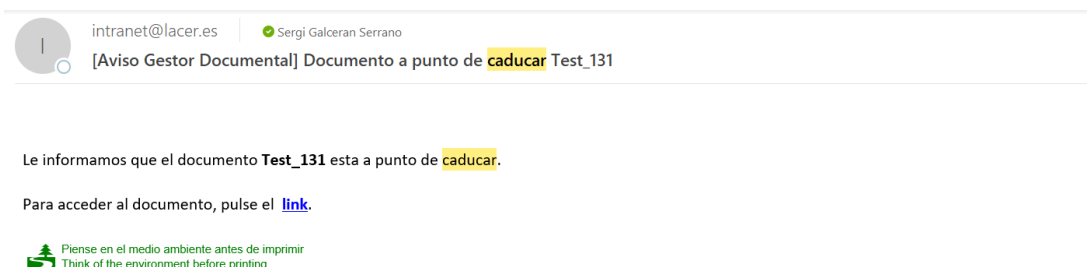


Ilustración 5.13 Correo caducidad documento

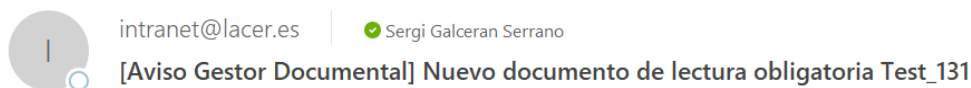
En el correo que se envía se indica el nombre del documento que va a caducar en el título y en el cuerpo y se añade un link que nos lleva al documento.

### 5.5 Leídos y comprendidos

Al crear un documento controlado, los usuarios que se añadan en el campo del formulario "Personas leído y comprendido" se les envía un correo avisando que tienen pendiente un documento por aceptar. El texto que se envía en el correo



se obtiene también de la lista de configuración de correos. Al igual que pasaba con los recordatorios aparece el título del documento en el título y en el cuerpo del correo. Si el usuario hace clic en el enlace, este le redirigirá al formulario (Ilustración 5.15) donde tendrá que aceptar el leído y comprendido del documento. Otra manera de acceder al formulario es haciendo clic en el nombre del documento en la biblioteca de documentos controlados.




Por favor, lea el documento **Test\_131** que se encuentra en el [link](#) y confirme su lectura y comprensión.



### Ilustración 5.14 Correo nuevo documento leído y comprendido

Para poder marcar la casilla de Leído y comprendido el usuario deberá hacer clic primero en el nombre del documento para que este se abra. Una vez se haya abierto el documento el usuario ya podrá hacer clic. Hasta que el usuario no haya abierto el documento la casilla de leído y comprendido no será clicable.

 Por favor lea el presente documento y a continuación marque la opción "Leído y comprendido"

Nombre del documento [689409-146](#) 📄

Código 19-ES-INS-CAL

Departamento Calidad

Tipo de Documento Instrucciones

Idioma Español

Empresa ICCSA

Promotor ☐ Oscar Salobreña

Fecha de publicación 12/06/2019 17:51

Vigente Sí

☐ Leído y comprendido

Tipo de contenido: Controlados

Creado el 12/06/2019 17:50 por ☐ Oscar Salobreña

Última modificación realizada el 12/06/2019 17:51 por ☐ Oscar Salobreña

Figura 5.15 Formulario leído y comprendido sin aceptar

Una vez se ha aceptado, si el usuario vuelve a entrar al formulario aparecerá otro diferente (Ilustración 5.16) informándole que ese documento ya ha sido leído y comprendido.

**Usted ya ha leído y prestado conformidad al presente documento**

Nombre del documento **689409-146** 📄

Código **19-ES-INS-CAL**

Departamento **Calidad**

Tipo de Documento **Instrucciones**

Idioma **Español**

Empresa **ICCSA**

Promotor ☐ Oscar Salobreña

Fecha de publicación **12/06/2019 17:51**

Vigente **Sí**

☒ **Leído y comprendido**

Tipo de contenido: Controlados

Creado el 12/06/2019 17:50 por ☐ Oscar Salobreña

Última modificación realizada el 12/06/2019 17:51 por ☐ Oscar Salobreña

Cerrar

**Ilustración 5.16 Formulario leído y comprendido aceptado**

Una vez el usuario hace clic y cierra el formulario, esta información se almacena en una lista de estados (Ilustración 5.17), donde los usuarios autorizados pueden ver aquellos usuarios que han leído y comprendido el documento y aquellos que están pendientes. También nos indica si se les ha enviado algún recordatorio.

#### Documentación Oficial Compañía - Estado Documentos ⓘ

[+ nuevo elemento](#) o [modificar](#) esta lista

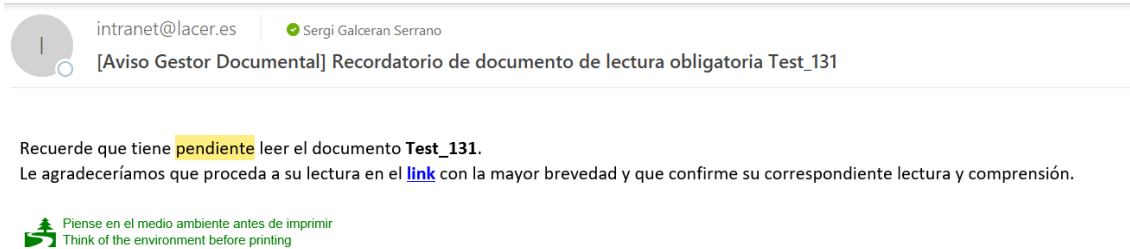
Mi peticiones   Pendientes de Aceptar   **Todos los elementos**   ...  

✓	Nombre del documento	Correo personas a aceptar	Nombre personas a aceptar	Departamento	Fecha aceptación	Fecha de envío	Estado	Número de recordatorios	Fecha último recordatorio
	Test353	<a href="#">lucas.guzman@raona.com</a>	Beatriz Buisac	Compliance		12/06/2019 23:37	Pendiente	0	
	689409-146	<a href="#">sergi.galceran@raona.com</a>	Oscar Salobreña	Calidad		12/06/2019 17:51	Leído y comprendido	0	

**Ilustración 5.17 Lista estado documentos**

En la lista de configuración contiene el número de recordatorios máximos que se enviarán y también cada cuantos días se enviara un nuevo recordatorio. En

caso que hayan transcurrido los días que marca la lista se envía un correo a la persona avisando que tiene pendiente de marcar como leído y comprendido un documento. Al igual que sucede con los casos anteriores el texto del correo se obtiene de la lista de configuración de correos.



### Ilustración 5.18 Correo recordatorio lectura documento

En el correo que se envía (Ilustración 5.18), se puede observar que sigue la misma estructura que los anteriores. Se puede observar el nombre del documento en el título y en el cuerpo del correo y, un link que nos permite acceder directamente al formulario.

## **CAPÍTULO 6. CONCLUSIONES**

En este último apartado se detallarán unas conclusiones sobre el proyecto realizado.

Para concluir el trabajo, se puede decir que se han cumplido los objetivos que abarcaba el proyecto. Con la solución que se ha implementado en SharePoint, se ha conseguido que los documentos puedan ser consultados desde un único punto y, además les permite escoger el nivel de permisos para cada documento que era el segundo objetivo de este proyecto.

Además, la implementación de gestión de leídos y comprendidos ha permitido organizar de una manera ágil y sencilla el proceso de aceptación de los documentos de lectura obligatoria para los trabajadores de la compañía.

## BIBLIOGRAFIA

[1] Creación soluciones de SharePoint [online] <<https://docs.microsoft.com/es-es/visualstudio/sharepoint/create-sharepoint-solutions?view=vs-2019>>

[2] Caml Query [online] <<http://sharepoint-works.blogspot.com/2012/05/caml-query-tutorial-for-sharepoint.html>>